



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ - ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΑΣ (Τ.Π.Ε.) ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ**

**Η ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕΣΑ ΑΠΟ ΠΑΙΧΝΙΔΙΑ
(ΤΟ ΠΕΡΙΒΑΛΛΟΝ GREENFOOT)
(TEACHING OBJECT-ORIENTED PROGRAMMING
THROUGH GAMES – GREENFOOT ENVIRONMENT)**

**ΣΥΝΤΑΚΤΗΣ:
ΜΟΧΛΑ ΑΝΑΣΤΑΣΙΑ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ:
ΠΑΠΑΔΟΠΟΥΛΟΥ ΘΕΟΔΩΡΑ**

ΛΑΜΙΑ 2021



**UNIVERSITY OF THESSALY – SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS WITH
APPLICATIONS IN BIOMEDICINE
POSTGRADUATE STUDY PROGRAM
COMPUTER SCIENCE AND ICT IN EDUCATION**

**TEACHING OBJECT-ORIENTED PROGRAMMING
THROUGH GAMES - (GREENFOOT ENVIRONMENT)**

AUTHOR:
MOCHLA ANASTASIA

**MASTER THESIS
SUPERVISOR PROFESSOR:**
PAPADOPOULOU THEODORA

LAMIA 2021

«Υπεύθυνη Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης»

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, και γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα και ενυπογράφως ότι η παρούσα εργασία με τίτλο [«Η διδασκαλία του αντικειμενοστραφούς προγραμματισμού μέσα από παιχνίδια (Το περιβάλλον Greenfoot)»] αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές από τις οποίες χρησιμοποίησα δεδομένα, ιδέες, φράσεις, προτάσεις ή λέξεις, είτε επακριβώς (όπως υπάρχουν στο πρωτότυπο ή μεταφρασμένες) είτε με παράφραση, έχουν δηλωθεί κατάλληλα και ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η ΔΗΛΩΝ/-ΟΥΣΑ

Ημερομηνία

Υπογραφή

Η ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕΣΑ ΑΠΟ ΠΑΙΧΝΙΔΙΑ (ΤΟ ΠΕΡΙΒΑΛΛΟΝ GREENFOOT)

ΜΟΧΛΑ ΑΝΑΣΤΑΣΙΑ

Τριμελής Επιτροπή:

Ονοματεπώνυμο,(επιβλέπων/σα)

Ονοματεπώνυμο,

Ονοματεπώνυμο,

Επιστημονικός Σύμβουλος:

Ονοματεπώνυμο.....

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία έχει ως θέμα τη διδασκαλία του αντικειμενοστραφούς προγραμματισμού με τη χρήση παιχνιδιών, εστιάζοντας στο εκπαιδευτικό περιβάλλον Greenfoot. Στόχος της είναι να διαπιστώσει το βαθμό, στον οποίο τα παιχνίδια βοηθούν στην κατανόηση και εκμάθηση του προγραμματισμού και καθιστούν οικείες και προσιτές τις σχετικές έννοιες.

Καθώς ο προγραμματισμός θεωρείται μία σύνθετη και πολύπλευρη διαδικασία και συχνά ο αρχάριος προγραμματιστής αποθαρρύνεται στην προσπάθειά του να κατανοήσει βασικές προγραμματιστικές έννοιες, αλλά και να τις χρησιμοποιήσει στην επίλυση προβλημάτων, προέκυψε ως επιτακτική ανάγκη η αναζήτηση νέων τεχνικών και προσεγγίσεων διδασκαλίας. Η εκπαιδευτική διαδικασία εμπλουτίστηκε με υπολογιστικές και δικτυακές τεχνολογίες, οι οποίες δημιούργησαν νέες συνθήκες μάθησης και επέφεραν σημαντικές αλλαγές στο εκπαιδευτικό σύστημα της χώρας.

Στο πλαίσιο αυτό, τα ψηφιακά παιχνίδια θεωρήθηκαν ελκυστικός τρόπος προσέγγισης των νέων, μιας και είναι ιδιαίτερα δημοφιλή σε αυτούς. Κατά συνέπεια, κρίθηκε απαραίτητο να ελεγχθεί κατά πόσο αν χρησιμοποιηθούν σωστά κατά τη διδασκαλία τα ψηφιακά παιχνίδια, είναι πιθανό να οδηγήσουν στην επίτευξη των μαθησιακών στόχων.

ABSTRACT

This paper deals with the teaching of object-oriented programming using games, focusing on the learning environment Greenfoot. It aims to determine the extent in which games help to understand and learn programming and make its related concepts familiar and understandable.

As programming is considered a complicated and multifaceted process and often the novice developer is discouraged from trying to understand basic programming concepts, but also to use them in problem solving, it emerged as imperative need to search for new teaching techniques and approaches. The educational process was enriched with computer and network technologies, which created new conditions for learning and led to significant changes in the country's educational system.

In this context, digital games have been considered an attractive way of approaching young people, as they are very popular to them. Therefore, it was necessary to examine whether digital games are likely to lead to the achievement of learning objectives, if used properly in teaching.

Έννοιες & Λέξεις-κλειδιά: ΤΠΕ, Πληροφορική, διδακτική προγραμματισμού, αντικειμενοστραφής προγραμματισμός, παιχνίδια, ψηφιακά παιχνίδια, εκπαιδευτικό περιβάλλον Greenfoot, περιβάλλον εργασίας, εκπαιδευτικό σενάριο, έρευνα

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	5
ABSTRACT	6
ΠΕΡΙΕΧΟΜΕΝΑ.....	7
ΕΙΣΑΓΩΓΗ.....	10
ΜΕΡΟΣ Α: ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ	12
ΕΝΟΤΗΤΑ 1. ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΕΚΠΑΙΔΕΥΣΗ.....	12
1.1. Οι Τεχνολογίες Πληροφορίας και Επικοινωνίας (ΤΠΕ) στην εκπαίδευση	12
1.1.1. Ορισμός ΤΠΕ.....	12
1.1.2. Ένταξη και ενσωμάτωση των ΤΠΕ στην εκπαίδευση	13
1.1.3. Η συμβολή των ΤΠΕ στην εκπαίδευση.....	14
1.2. Η Διδασκαλία της Πληροφορικής	15
1.2.1. Ορισμός Πληροφορικής	15
1.2.2. Ένταξη της Πληροφορικής στην εκπαίδευση.....	16
1.3. Η Διδασκαλία του Προγραμματισμού	16
1.3.1. Ορισμός Προγραμματισμού	17
1.3.2. Προσεγγίσεις κατά τη διδασκαλία του Προγραμματισμού.....	18
1.3.3. Δυσκολίες κατά τον Προγραμματισμό	19
1.4. Προγραμματισμός και παιχνίδια	21
1.4.1. Ψηφιακά παιχνίδια	22
1.4.1.1. Ορισμός.....	22
1.4.1.2. Χαρακτηριστικά	23
1.4.1.3. Θετικές και αρνητικές επιδράσεις	24
1.4.1.4. Ένταξη στην εκπαιδευτική διαδικασία	26
1.4.2. Εκπαιδευτικά περιβάλλοντα – Σύγκριση.....	27

ΕΝΟΤΗΤΑ 2. ΕΚΠΑΙΔΕΥΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ GREENFOOT	34
2.1. Το προγραμματιστικό περιβάλλον Greenfoot.....	34
2.2. Ιστορία	35
2.3. Το προγραμματιστικό μοντέλο.....	37
2.4. Υποστήριξη.....	38
2.5. Εγκατάσταση του περιβάλλοντος Greenfoot.....	39
2.6. Περιβάλλον εργασίας.....	40
2.7. Παρουσίαση σεναρίου Greenfoot.....	42
ΜΕΡΟΣ Β: ΕΡΕΥΝΗΤΙΚΟ ΠΛΑΙΣΙΟ	57
ΕΙΣΑΓΩΓΗ	57
ΕΝΟΤΗΤΑ 3. ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ.....	59
3.1. Αντικειμενικός σκοπός έρευνας	59
3.2. Ερευνητική μέθοδος	59
3.3. Υποθέσεις έρευνας	59
3.4. Δείγμα έρευνας.....	60
3.5. Εργαλείο έρευνας.....	61
3.6. Ανάλυση αποτελεσμάτων	61
ΕΝΟΤΗΤΑ 4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΕΥΝΑΣ	62
4.1. Δημογραφικά στοιχεία.....	62
4.2. Στοιχεία γενικών γνώσεων	63
4.3. Στοιχεία κατανόησης αντικειμένου.....	66
4.4. Στοιχεία περιβάλλοντος.....	69
ΕΝΟΤΗΤΑ 5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΕΥΝΑΣ.....	73
ΕΝΟΤΗΤΑ 6. ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΓΑΣΙΑΣ.....	74
ΒΙΒΛΙΟΓΡΑΦΙΑ	76
ΕΛΛΗΝΟΓΛΩΣΣΗ	76
ΞΕΝΟΓΛΩΣΣΗ.....	79

ΔΙΚΤΥΑΚΟΙ ΤΟΠΟΙ.....	81
ΠΑΡΑΡΤΗΜΑ Ι – ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ.....	83
ΠΑΡΑΡΤΗΜΑ ΙΙ – ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ.....	101
ΠΑΡΑΡΤΗΜΑ ΙΙΙ – ΕΙΚΟΝΕΣ.....	105
ΠΑΡΑΡΤΗΜΑ ΙV – ΠΙΝΑΚΕΣ.....	106

ΕΙΣΑΓΩΓΗ

Στην εποχή μας, η ολοένα αυξανόμενη χρήση των νέων τεχνολογιών και μέσων επικοινωνίας τις έχει καταστήσει αναπόσπαστο κομμάτι της καθημερινότητας. Οι νέες τεχνολογίες εμπλέκονται και συμβάλλουν αποτελεσματικά σε όλες σχεδόν τις επιστήμες. Όπως ήταν φυσικό, νέες εξελίξεις δρομολογήθηκαν και στην εκπαιδευτική διαδικασία, οδηγώντας σε αναπροσαρμογή των μεθόδων και τρόπων διδασκαλίας με την αξιοποίηση των ΤΠΕ (Τεχνολογίες της Πληροφορίας και των Επικοινωνιών). Η εκπαιδευτική διαδικασία εμπλουτίστηκε με υπολογιστικές και δικτυακές τεχνολογίες, οι οποίες δημιούργησαν νέες συνθήκες μάθησης και επέφεραν σημαντικές αλλαγές στο εκπαιδευτικό σύστημα της χώρας.

Στις μέρες μας, είναι γεγονός ότι οι νέοι χρησιμοποιούν καθημερινά την ψηφιακή τεχνολογία μέσω των υπολογιστών, των «έξυπνων τηλεφώνων» (smart phones), των φορητών συσκευών (π.χ. tablet) κ.α. Οι τομείς, που δείχνουν να ελκύουν την προσοχή τους είναι η μουσική, η προβολή βίντεο, η λήψη φωτογραφιών, τα μέσα κοινωνικής δικτύωσης και κυρίως, η χρήση ηλεκτρονικών παιχνιδιών. Για τον λόγο αυτό, αναπτύσσονται συνεχώς συστήματα μάθησης, που βασίζονται στους παραπάνω τομείς με απώτερο σκοπό να κεντρίσουν το ενδιαφέρον των εκπαιδευόμενων και να μεταδώσουν τη γνώση με ουσιαστικό και βιωματικό τρόπο μέσω της αλληλεπίδρασης (interactivity). Αυτός ο τρόπος διδασκαλίας, όπου η επίτευξη των στόχων είναι άμεσα ορατή (π.χ. στην οθόνη του Η/Υ), διατηρεί αμείωτο το ενδιαφέρον του εκπαιδευόμενου και του παρέχει αίσθημα ευφορίας, σιγουριάς και ικανοποίησης, αφού η εκπαιδευτική διαδικασία καθίσταται μια διασκεδαστική δραστηριότητα με γρίφους, στόχους και αποτελέσματα.

Ειδικότερα, κατά τη διδασκαλία του Προγραμματισμού δίνεται η δυνατότητα της καλύτερης δυνατής χρήσης των νέων εκπαιδευτικών μεθόδων. Μέχρι σήμερα, πολλές μέθοδοι έχουν προταθεί και υλοποιηθεί, όπως για παράδειγμα αυτές των έξυπνων πρακτόρων (smart agents), των αφηγήσεων με βίντεο και ήχο, της τυποποιημένης συμπλήρωσης, της σύνθεσης με μπλοκ βασικών δομών κτλ. Ωστόσο, τις μεθόδους αυτές έρχεται να ανταγωνιστεί η μάθηση μέσα από παιχνίδια, τα οποία αποτελούν αναπόσπαστο τμήμα της ζωής των νέων και καταφέρνουν να τους ελκύουν με τις όμορφες εικόνες, τους ήχους και τη σχεδιοκίνηση (animation).

Η παρούσα διπλωματική εργασία, χρησιμοποιώντας το περιβάλλον προγραμματισμού Greenfoot, έχει ως στόχο να αναδείξει την αποτελεσματικότητα της χρήσης των παιχνιδιών στη διαδικασία εκμάθησης του Προγραμματισμού. Αν, δηλαδή, μέσα από τα παιχνίδια, οι όροι, οι έννοιες και οι λειτουργίες παρουσιάζονται με τέτοιο τρόπο, ώστε να διευκολύνεται η κατανόηση, η εκμάθηση και η εμπέδωση της διαδικασίας, που ονομάζεται Προγραμματισμός.

Το πρώτο μέρος της παρούσας εργασίας είναι θεωρητικό. Στην πρώτη ενότητα επιχειρείται η παρουσίαση του ρόλου των ΤΠΕ στην εκπαίδευση, δίνοντας έμφαση στην ένταξη και ενσωμάτωσή τους στην εκπαίδευση, καθώς, επίσης και στη συμβολή τους σε αυτή. Στη συνέχεια, γίνεται αναφορά στη διδασκαλία του μαθήματος της Πληροφορικής και στον τρόπο με τον οποίο έχει ενταχθεί στην εκπαίδευση. Παράλληλα, παρουσιάζεται η διδασκαλία του προγραμματισμού, εστιάζοντας στα προβλήματα, που αυτή συναντά και στην προσπάθεια επίλυσης αυτών των προβλημάτων με τη χρήση των παιχνιδιών. Η πρώτη ενότητα του θεωρητικού μέρους ολοκληρώνεται με την παρουσίαση εφαρμογών και εκπαιδευτικών περιβάλλοντων, τα οποία θα μπορούσαν να χρησιμοποιηθούν στην εκπαίδευση. Στη δεύτερη ενότητα γίνεται παρουσίαση του εκπαιδευτικού περιβάλλοντος Greenfoot και συγχρόνως δίνεται συνοπτικά ο τρόπος εγκατάστασής του, η περιγραφή και επεξήγηση του περιβάλλοντος εργασίας, αλλά και η διασαφήνιση βασικών εντολών και συναρτήσεων.

Το δεύτερο μέρος της παρούσας εργασίας αφορά στην έρευνα. Το εκπαιδευτικό περιβάλλον Greenfoot χρησιμοποιήθηκε για τον μετασχηματισμό ημιτελούς παιχνιδιού, το οποίο, αφού διδάχτηκε σε σχολικές μονάδες, τέθηκε στην κρίση των εκπαιδευόμενων μέσα από ένα ερωτηματολόγιο 15 ερωτήσεων. Στόχος ήταν να καταγραφούν οι γνώσεις, που αποκόμισαν οι εκπαιδευόμενοι και οι απόψεις τους για τη λειτουργία και χρήση του εν λόγω περιβάλλοντος, ώστε να διαπιστωθεί η συμβολή του στην κατανόηση και εκμάθηση του προγραμματισμού.

ΜΕΡΟΣ Α: ΘΕΩΡΗΤΙΚΟ ΠΛΑΙΣΙΟ

ΕΝΟΤΗΤΑ 1. ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΕΚΠΑΙΔΕΥΣΗ

1.1. Οι Τεχνολογίες Πληροφορίας και Επικοινωνίας (ΤΠΕ) στην εκπαίδευση

Στην ενότητα αυτή, γίνεται προσπάθεια κατανόησης της έννοιας «Τεχνολογίες της Πληροφορίας και της Επικοινωνίας (ΤΠΕ)», παραθέτοντας ορισμούς και απόψεις, όπως διατυπώθηκαν από Έλληνες και ξένους επιστήμονες. Παράλληλα, παρατίθενται οι επικρατέστερες προσεγγίσεις για την αξιοποίηση και ένταξη των ΤΠΕ στη διδακτική διαδικασία και αναλύεται η συμβολή των ΤΠΕ στην εκπαίδευση μέσα από έρευνες, που έχουν πραγματοποιηθεί, προκειμένου να αποσαφηνιστεί ο τρόπος, με τον οποίο επηρεάζουν οι τεχνολογίες αυτές τόσο την εξέλιξη του εκπαιδευόμενου, όσο και τη βελτίωση του εκπαιδευτικού.

1.1.1. Ορισμός ΤΠΕ

Καθώς οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας (ΤΠΕ) αποτέλεσαν σημαντικό παράγοντα στην καθημερινότητα του ανθρώπου και έγιναν αναπόσπαστο κομμάτι σχεδόν όλων των δραστηριοτήτων του (Prensky, 2004), έγινε επιτακτική η ανάγκη διατύπωσης του ορισμού τους. Αρκετοί ερευνητές (Κόμης, Σολομωνίδου κ.α.) προχώρησαν στη διατύπωση ορισμών, προκειμένου να αναλυθεί και να αποσαφηνιστεί ο όρος.

Έξω από το πλαίσιο της εκπαίδευσης, οι ΤΠΕ είναι το σύνολο των επαγγελματικών χώρων, οι οποίοι σχετίζονται με τη μελέτη, σχεδίαση, ανάπτυξη, υλοποίηση, συντήρηση και διαχείριση υπολογιστικών πληροφοριακών συστημάτων, κυρίως όσον αφορά εφαρμογές λογισμικού και υλικού υπολογιστών ([wikipedia/Τεχνολογία πληροφοριών](https://el.wikipedia.org/wiki/Τεχνολογία_πληροφοριών)).

Σε μια διαφορετική διατύπωση, οι ΤΠΕ περιλαμβάνουν όλα τα ψηφιακά μέσα, που μπορούν να υποστηρίξουν μία σειρά ενεργειών, οι οποίες αφορούν στην πληροφορία – π.χ. αναζήτηση, επεξεργασία, διαμόρφωση, συλλογή, ανάλυση, παρουσίαση κ.τ.λ. - ([Διάλεξη/ΤΠΕ στη σχολική τάξη](#)) και έτσι δύνανται να επιλύσουν προβλήματα και να συμβάλουν στην εξέλιξη της ζωής (Σολομωνίδου, Σταυρίδου, 1994, Τάσση, 2014).

Ένας άλλος ορισμός αναφέρει ότι ο όρος Τεχνολογίες της Πληροφορίας και της Επικοινωνίας (ΤΠΕ) χρησιμοποιείται, για να χαρακτηρίσει «τις τεχνολογίες εκείνες, που επιτρέπουν την επεξεργασία και τη μετάδοση μιας ποικιλίας μορφών αναπαράστασης της πληροφορίας (σύμβολα, εικόνες ήχοι, βίντεο) και αφετέρου τα μέσα, που είναι φορείς αυτών των άυλων μηνυμάτων» (Κόμης, 2004).

Σε κάθε περίπτωση, οι ΤΠΕ έχουν ως επίκεντρο τον Η/Υ (Σολομωνίδου, 2007) και συγκεντρώνουν όλα τα στοιχεία και χαρακτηριστικά, που απαιτούνται, για να κατοχυρώσουν μία σημαντική θέση στην εκπαιδευτική διαδικασία, να ταυτιστούν με αυτή και να αποτελέσουν απαραίτητο εργαλείο της.

1.1.2. Ένταξη και ενσωμάτωση των ΤΠΕ στην εκπαίδευση

Οι ραγδαίες τεχνολογικές εξελίξεις κατέστησαν επιτακτική ανάγκη την αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας σχεδόν σε κάθε ανθρώπινη δραστηριότητα. Υπό αυτές τις συνθήκες δεν ήταν δυνατό να μείνει ανεπηρέαστη η εκπαιδευτική διαδικασία.

Οι επικρατέστερες προσεγγίσεις για την αξιοποίηση και ένταξη των ΤΠΕ στη διδακτική διαδικασία είναι τρεις (Μακράκης & Κοντογιαννοπούλου-Πολυδωρίδη, 1995; Κόμης, 2004).

Η **Τεχνοκεντρική ή κάθετη προσέγγιση** (συναντάται στη βιβλιογραφία και ως **Τεχνοκεντρικό μοντέλο ή πρότυπο**) αντιμετωπίζει τις ΤΠΕ ως αυτόνομο γνωστικό αντικείμενο και τις ταυτίζει με την Πληροφορική. Κατά συνέπεια, βασικός στόχος είναι ο τεχνολογικός αλφαριθμητισμός, που αφορά τόσο στη χρήση του Η/Υ, όσο και στην ανάπτυξη δεξιοτήτων προγραμματισμού. Ως αυτόνομο γνωστικό αντικείμενο έχει συγκεκριμένους σκοπούς και στόχους, τρόπους διδασκαλίας και διδακτικής, οι οποίοι παρατίθενται στο αναλυτικό πρόγραμμα.

Η **Ολιστική ή Ολοκληρωμένη προσέγγιση** (συναντάται στη βιβλιογραφία και ως **Ολιστικό μοντέλο ή πρότυπο**) αναφέρεται στις ΤΠΕ ως εργαλείο στήριξης της εκπαιδευτικής διαδικασίας στο σύνολο των γνωστικών αντικειμένων. Οι ΤΠΕ χρησιμοποιούνται ως εργαλεία γνώσης, έρευνας και μάθησης σε κάθε γνωστικό αντικείμενο, ενώ η χρήση του υπολογιστή και η διδασκαλία αυτής πραγματώνεται κατά τη διδασκαλία άλλων αντικειμένων και όχι στο πλαίσιο ενός αυτόνομου μαθήματος Πληροφορικής.

Τέλος, η **Πραγματολογική ή Εφικτή-μεικτή προσέγγιση** (συναντάται στη βιβλιογραφία και ως **Πραγματολογικό ή Μεικτό μοντέλο ή πρότυπο**) συνδυάζει τις δύο προηγούμενες προσεγγίσεις, αποτελώντας παράλληλα το μεταβατικό στάδιο από το τεχνοκεντρικό στο ολιστικό μοντέλο. Επομένως, περιλαμβάνει την ύπαρξη ενός αυτόνομου γνωστικού αντικειμένου (μάθημα Πληροφορικής), ενώ παράλληλα αξιοποιεί και χρησιμοποιεί τις ΤΠΕ ως μέσο στήριξης της διδασκαλίας όλων των γνωστικών αντικειμένων. Με τον τρόπο αυτό, οι ΤΠΕ είναι αυτόνομο γνωστικό αντικείμενο, αλλά και εργαλείο γνώσης, έρευνας και μάθησης όλων των γνωστικών αντικειμένων (Ράπτης και Ράπτη 2002).

1.1.3. Η συμβολή των ΤΠΕ στην εκπαίδευση

Η εισαγωγή, ένταξη και ενσωμάτωση των ΤΠΕ στην εκπαίδευση βελτιώνει την ποιότητα της μάθησης και αναβαθμίζει την εκπαιδευτική διαδικασία. Οι ΤΠΕ απαρτίζονται από πληθώρα εργαλείων, όπως ο Η/Υ, η δικτύωση, τα εκπαιδευτικά και μη λογισμικά, τα σύγχρονα διαδραστικά εκπαιδευτικά συστήματα (interactivity), τα τηλεπικοινωνιακά μέσα κ.α., τα οποία δύνανται να υποστηρίξουν τη διδασκαλία και τη μάθηση, εφόσον αξιοποιηθούν σωστά.

Παράλληλα, προσφέρουν άμεση πρόσβαση σε μεγάλο όγκο πληροφοριών και δεδομένων, δίνουν τη δυνατότητα παρουσίασης αυτών των δεδομένων, προάγουν τη συνεργασία, εξελίσσουν την επικοινωνία, ενώ συμβάλλουν στην καλλιέργεια και ανάπτυξη δεξιοτήτων. Αυτά τα ιδιαίτερα στοιχεία, που χαρακτηρίζουν τις ΤΠΕ, είναι που τις καθιστούν αποτελεσματικό εργαλείο στην εκπαιδευτική διαδικασία και τη μάθηση (Βοσνιάδου, 2006).

Μέσα από έρευνες διαπιστώνεται ότι με τη χρήση των ΤΠΕ στη διδασκαλία παρατηρείται μεγαλύτερη προσήλωση, εντονότερο ενδιαφέρον και αυξημένη συμμετοχή των εκπαιδευόμενων χάρη στην αυτενέργεια, την αναζήτηση και ανακάλυψη και την αλληλεπίδραση μεταξύ τους (Αναστασιάδης κ.α., 2010). Ως αποτέλεσμα η γνώση αφομοιώνεται ταχύτερα και ευκολότερα και οι εκπαιδευόμενοι καλλιεργούν την κριτική σκέψη και αποκτούν πρόσβαση στη νέα γνώση (Δημητρακοπούλου, 2004).

Συγχρόνως, οι εκπαιδευόμενοι δείχνουν ενδιαφέρον και μελετούν περισσότερο, καθώς αποφασίζουν οι ίδιοι για το είδος γνώσης, που επιθυμούν να λάβουν (Μικρόπουλος, 2006), γεγονός που καθιστά τη μελέτη ουσιαστική, αλλά και ευχάριστη.

Γενικότερα, αποκτούν κοινωνικές δεξιότητες μέσα από τη συνεργασία και την αλληλεπίδραση, είτε δια ζώσης, είτε μέσω δικτύων, εκπαιδεύονται στην επίλυση προβλημάτων, διαμορφώνουν την εξέλιξη της μάθησης και εξοικειώνονται με την έννοια της δια βίου μάθησης, θέτοντας βάσεις για το μέλλον. (Καλούρη-Αντωνοπούλου & Σιγάλας, 2006).

Εξίσου θετική, όμως, είναι η επίδραση των ΤΠΕ και στον εκπαιδευτικό, ο οποίος αναλαμβάνει πλέον τον ρόλο του βοηθού, συνεργάτη και καθοδηγητή. Ο εκπαιδευτικός δίνει τις γενικές κατευθύνσεις, τονίζει τα βασικά σημεία και ενθαρρύνει τους εκπαιδευόμενους, ώστε να αναλάβουν πρωτοβουλίες. (Καράμηνas, 2006; Κατωπόδης, 2010; Μητροπούλου, 2008; Λαφατζή, 2005).

Ο εκπαιδευτικός καλείται να προσαρμόσει ανάλογα την οργάνωση και διεξαγωγή του μαθήματος, να παραμερίσει τυχόν προκαταλήψεις και κωλύματα και να εναρμονιστεί με τις νέες συνθήκες, που παρέχει η χρήση ΤΠΕ. Έτσι, θα κατορθώσει να βελτιώσει την εκπαιδευτική διαδικασία, να ερμηνεύσει σφαιρικά τα αποτελέσματά της και να τελέσει επιτυχώς τα διδακτικά του καθήκοντα. (Λαφατζή, 2005; Ράπτης & Ράπτη, 1998). Άλλωστε, σύμφωνα με τους Καλούρη-Αντωνοπούλου &

Σιγάλα (2006), ο ρόλος του είναι καθοριστικός και απαραίτητος για τη μάθηση, διότι χωρίς την παρουσία του ο εκπαιδευόμενος αποξενώνεται από το περιβάλλον του και απομονώνεται, χάνοντας όλα τα οφέλη των ΤΠΕ.

Συμπερασματικά, η ένταξη των ΤΠΕ στην εκπαίδευση δύναται να αλλάξει και να εξελίξει θετικά τη μαθησιακή διαδικασία, εξελίσσοντας και βελτιώνοντας ταυτόχρονα εκπαιδευόμενους και εκπαιδευτικούς, ενισχύοντας την κριτική σκέψη των μεν και προκαλώντας την ανοιχτή ματιά των δε.

1.2. Η Διδασκαλία της Πληροφορικής

Η ενότητα αυτή αναφέρεται στη διδασκαλία του αντικειμένου της Πληροφορικής. Αναλύεται το γνωστικό αντικείμενο και οι εκπαιδευτικοί στόχοι και σκοποί του μαθήματος. Στη συνέχεια, ακολουθεί μια σύντομη αναφορά στον τρόπο και χρόνο ένταξης της Πληροφορικής ως γνωστικό αντικείμενο στην εκπαιδευτική διαδικασία.

1.2.1. Ορισμός Πληροφορικής

Με την πάροδο των ετών, στη χώρα μας, αναγνωρίζονται ενέργειες για τη χρήση των ΤΠΕ σε όλα τα γνωστικά αντικείμενα. Ωστόσο, σε μεγάλο βαθμό οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας προσεγγίζονται ακόμα κυρίως τεχνοκεντρικά, ως αυτόνομο γνωστικό εργαλείο μέσω του μαθήματος της Πληροφορικής.

Η επιστήμη της Πληροφορικής ασχολείται με τον σχεδιασμό, τη δημιουργία, τον έλεγχο και τη χρήση των συστημάτων εκείνων, που αφορούν στη μετάδοση, αποθήκευση και επεξεργασία της ψηφιακής πληροφορίας (Anderson & Van Weert, 2002).

Ως γνωστικό αντικείμενο, στο πλαίσιο της εκπαιδευτικής διαδικασίας, η Πληροφορική έχει ως στόχο να παρέχει γνώσεις, ώστε να γίνουν κατανοητές και διαχρονικές οι βασικές έννοιές της, να καλλιεργήσει δεξιότητες σχετικές με την επίλυση προβλημάτων μέσω περιβαλλόντων προγραμματισμού και διαφόρων λογισμικών και τέλος να οδηγήσει στην απόκτηση βασικών ικανοτήτων ως προς τη χρήση λειτουργικών περιβάλλοντων, την επεξεργασία κειμένων, τη διαχείριση υπολογιστικών φύλλων, την ανάπτυξη βάσεων δεδομένων, τη δημιουργία παρουσιάσεων, τη χρήση του Διαδικτύου, κ.α.

Ειδικότερα, στο πλαίσιο διδασκαλίας της πληροφορικής ο εκπαιδευόμενος καλείται να κατανοήσει και να αφομοιώσει βασικές έννοιες και όρους της Πληροφορικής (hardware-υλικό, software-λογισμικό, δομές και μεθόδους επεξεργασίας της πληροφορίας, προγραμματιστικές δομές κ.α.), να αποκτήσει ικανότητες μεθοδολογικού χαρακτήρα και δεξιότητες ως προς τη χρήση των

Τεχνολογιών Πληροφορίας και Επικοινωνίας (ΤΠΕ), να αναπτύξει αναλυτική, συνθετική και υπολογιστική σκέψη (δομημένη), επιλύοντας προβλήματα σε προγραμματιστικό περιβάλλον (αλγοριθμική προσέγγιση), να αξιολογήσει (κριτική σκέψη) και να αξιοποιήσει τις δυνατότητες, που του παρέχονται.

1.2.2. Ένταξη της Πληροφορικής στην εκπαίδευση

Στη χώρα μας, το μάθημα της Πληροφορικής (Μπράτιτσης, 2013) εισήχθηκε αρχικά στα Τεχνικά και Πολυκλαδικά Λύκεια στα μέσα της δεκαετίας του '80. Περίπου δέκα χρόνια μετά, αποτέλεσε αυτόνομο γνωστικό αντικείμενο του αναλυτικού Προγράμματος του Γυμνασίου χωρίς, όμως, αυτό να συνεπάγεται την άμεση και ταυτόχρονη εισαγωγή σε όλα τα Γυμνάσια της χώρας. Στα τέλη της δεκαετίας του '90, ξεκίνησε η εισαγωγή του στη βαθμίδα του Λυκείου, όπου, επίσης, χρειάστηκε η έλευση αρκετών ετών, έως ότου εισαχθεί σε κάθε Λύκειο της χώρας. Παράλληλα, η εισαγωγή του μαθήματος της Πληροφορικής ήταν άμεση και αφορούσε όλες σχεδόν τις ειδικότητες των Δημόσιων Ινστιτούτων Επαγγελματικής Κατάρτισης (ΔΙΕΚ), τα οποία εντάσσονται στο πλαίσιο της μεταδευτεροβάθμιας εκπαίδευσης ([Γενική Γραμματεία Δία Βίου Μάθησης](#)).

Η καθολική εισαγωγή του μαθήματος της Πληροφορικής στο ελληνικό εκπαιδευτικό σύστημα επήλθε με την εισαγωγή του μαθήματος στην Πρωτοβάθμια Εκπαίδευση. Αρχικά, εισήχθη πιλοτικά σε ορισμένα σχολεία και σταδιακά επεκτάθηκε σε όλα, καθώς εξοπλίστηκαν με υπολογιστές (Κόμης, 2004).

Ουσιαστικά, όμως, η Πληροφορική αναγνωρίστηκε ως γνωστικό αντικείμενο ίσης αξίας με άλλα βασικά και ιδιαιτέρως σημαντικά αντικείμενα (Γλώσσα, Μαθηματικά, Μελέτη Περιβάλλοντος, Δημιουργία και Έκφραση) μόλις το 2003 στο Διαθεματικό Ενιαίο Πλαίσιο Προγραμμάτων Σπουδών (ΔΕΠΠΣ, 2003).

1.3. Η Διδασκαλία του Προγραμματισμού

Με την ένταξη του μαθήματος του Προγραμματισμού στην εκπαιδευτική διαδικασία χρησιμοποιήθηκαν και εφαρμόστηκαν διάφορες προσεγγίσεις, προκειμένου να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα τόσο για τους εκπαιδευόμενους, όσο και για το αντικείμενο διδασκαλίας. Παράλληλα, καθοριστικό ρόλο για τη βελτίωση της εκπαιδευτικής διαδικασίας διαδραμάτισε ο εντοπισμός και στη συνέχεια η κατηγοριοποίηση όλων των δυσκολιών, που δύναται να συναντήσει ο εκπαιδευόμενος κατά την εκμάθηση του Προγραμματισμού.

1.3.2. Προσεγγίσεις κατά τη διδασκαλία του Προγραμματισμού

Κατά την ένταξη της εκμάθησης του προγραμματισμού στην εκπαιδευτική διαδικασία αναζητήθηκαν και εφαρμόστηκαν διαφορετικοί τρόποι και μοντέλα προσέγγισης και διδασκαλίας, με βάση τα υπάρχοντα μοντέλα διδασκαλίας και εκπαίδευσης.

Στα πρώτα στάδια της διδασκαλίας του προγραμματισμού χρησιμοποιήθηκαν κυρίως **εμπειρικές-τεχνοκεντρικές προσεγγίσεις**. Με βάση το συμπεριφοριστικό μοντέλο διδασκαλίας, διαμορφώθηκε ένα ανάλογο πλαίσιο διδασκαλίας, σύμφωνα με το οποίο ο εκπαιδευτής έχει τον κυρίαρχο ρόλο στη σχέση εκπαιδευόμενου-εκπαιδευτή. Ο εκπαιδευτής οφείλει να διδάξει το αντικείμενο εκμάθησης στους εκπαιδευόμενους, δηλαδή ορισμένες γλώσσες προγραμματισμού και τα χαρακτηριστικά τους (δομή, εντολές, σύνταξη κ.λπ). Οι εκπαιδευόμενοι καλούνται να αφομοιώσουν τις πληροφορίες και να τις εφαρμόσουν στο αντίστοιχο προγραμματιστικό περιβάλλον.

Παρόλο που αυτό το μοντέλο διδασκαλίας χρησιμοποιήθηκε ευρύτατα τόσο στο αντικείμενο του προγραμματισμού, όσο και σε άλλα αντικείμενα, σύντομα έφερε τους εκπαιδευτικούς σε αδιέξοδο, καθώς διαπιστώθηκε τόσο μέσα από την εκπαιδευτική, όσο και μέσα από την ερευνητική διαδικασία ότι δεν φέρει το επιθυμητό αποτέλεσμα. Αρχικά, η αποτελεσματικότητα της εκπαιδευτικής διαδικασίας εξαρτάται σε μεγάλο βαθμό από την κατάρτιση του εκπαιδευτή, καθώς είναι η μοναδική πηγή γνώσεων, καθιστώντας τον εκπαιδευόμενο παθητικό αποδέκτη. Επιπλέον, οι εκπαιδευόμενοι περιορίζονται συχνά στην απόκτηση στείων γνώσεων, τις οποίες αδυνατούν να εφαρμόσουν.

Η αναζήτηση και εύρεση ενός μοντέλου διδασκαλίας, που θα βοηθούσε τον εκπαιδευτικό να μεταδώσει αποτελεσματικά τις γνώσεις του, αλλά και τον εκπαιδευόμενο να κατανοήσει ουσιαστικά τις έννοιες του προγραμματισμού, ώστε να μπορεί να τις χρησιμοποιήσει στην επίλυση προβλημάτων, αποτέλεσε επιτακτική ανάγκη. Ως αποτέλεσμα αυτών ακολούθησαν διαφορετικές προσεγγίσεις διδασκαλίας. Μία από αυτές είναι η **ανακαλυπτική προσέγγιση** του προγραμματισμού, η οποία εστιάζει στο σχεδιασμό εκείνων των εκπαιδευτικών περιβάλλοντων, τα οποία, χρησιμοποιώντας ποικίλες τεχνολογίες οπτικοποίησης, προγραμματισμού και αλληλεπίδρασης, βοηθούν και ενισχύουν τους αρχάριους προγραμματιστές να δημιουργήσουν προγράμματα, να κατανοήσουν τη δυναμική τους συμπεριφορά, να εντοπίσουν τυχόν παρερμηνείες και σφάλματα, που σχετίζονται με αυτά και να δουν το αποτέλεσμα στον υπολογιστή τους (Ramadhan, 2001).

Παράλληλα, για τη διδασκαλία του προγραμματισμού χρησιμοποιήθηκε το **εποικοδομιστικό μοντέλο**, το οποίο υποστηρίζει ότι η αποτελεσματικότητα της διδασκαλίας είναι εφικτή μόνο όταν ο εκπαιδευτής λάβει υπόψη τον τρόπο, με τον οποίο οικοδομεί τις γνώσεις του ο εκπαιδευόμενος (Κόμης, 2001). Σύμφωνα με τους υποστηρικτές αυτού του μοντέλου, ο εκπαιδευόμενος βρίσκεται στο κέντρο της μάθησης, συμμετέχει στη μαθησιακή διαδικασία και οργανώνει τις γνώσεις του, όχι με

γραμματικό τρόπο, αλλά οικοδομώντας τις επάνω σε προηγούμενες γνώσεις. Το μοντέλο αυτό ενισχύθηκε σταδιακά με εκπαιδευτικές διαδικασίες βασισμένες στη συνεργατική μάθηση (Γρηγοριάδου κ.α. 2004) καθώς η αλληλεπίδραση και η συνεργασία μεταξύ των εκπαιδευόμενων συμβάλλουν καθοριστικά στη μάθηση του προγραμματισμού. (Williams & Kessler, 2000).

Συμπερασματικά, οι προσεγγίσεις, που θέτουν τον εκπαιδευόμενο στο κέντρο της εκπαιδευτικής διαδικασίας, στοχεύουν στη διεύρυνση και εξέλιξη της γνώσης και προάγουν τη συνεργατική μάθηση, οδηγούν στο επιθυμητό αποτέλεσμα. Η εκπαιδευτική εμπειρία και η έρευνα έχουν αποδείξει ότι, συνδυάζοντας την εξαιρετική γνώση του αντικειμένου και τη γνώση των βασικών αρχών της παιδαγωγικής (Major, Kyriacou & Brereton, 2012), ο εκπαιδευτικός επιτυγχάνει την αποτελεσματική διδασκαλία του προγραμματισμού.

1.3.3. Δυσκολίες κατά τον Προγραμματισμό

Τα τελευταία χρόνια έχει διεξαχθεί σημαντικός αριθμός ερευνών σχετικά με τις δυσκολίες, που παρουσιάζονται κατά τη διδασκαλία και εκμάθηση του προγραμματισμού (Soloway & Spohrer 1989, Τζιμογιάννης & Γεωργίου 1998, Τζιμογιάννης & Κόμης 1999). Οι έρευνες εστιάζουν κυρίως στους εκπαιδευόμενους και αφορούν τόσο στην ικανότητα κατανόησης βασικών προγραμματιστικών εννοιών, όσο και στη δυνατότητα εφαρμογής της αποκτηθείσας γνώσης για την επίλυση προβλημάτων.

Στόχος των ερευνών είναι η ανάλυση των αιτιών, που προκαλούν δυσκολίες στους αρχάριους προγραμματιστές, προκειμένου να καταστούν κατανοητές, ώστε να αντιμετωπιστούν με τη χρήση των κατάλληλων εργαλείων και μέσων, να προκύψουν νέες εκπαιδευτικές και μαθησιακές προσεγγίσεις διδασκαλίας και να χρησιμοποιηθούν τα κατάλληλα εκπαιδευτικά περιβάλλοντα εκμάθησης προγραμματισμού.

Οι δυσκολίες εκμάθησης του προγραμματισμού κατηγοριοποιήθηκαν σε πέντε βασικές ομάδες (Du Boulay, 1989):

- **Orientation (Προσανατολισμός):** Η ομάδα αυτή εμπεριέχει τις δυσκολίες, που αντιμετωπίζει ο εκπαιδευόμενος στην προσπάθειά του να ανακαλύψει τι είναι ο προγραμματισμός, τι είδους προβλήματα μπορεί να επιλύσει και ποια τα οφέλη από την εκμάθησή του.
- **Notional Machine (Εννοιολογική μηχανή):** Περιλαμβάνει δυσκολίες, οι οποίες σχετίζονται με τη δυνατότητα του εκπαιδευόμενου να αντιληφθεί τις γενικές ιδιότητες της εννοιολογικής μηχανής, που καλείται να ελέγξει, και τη σχέση της με τη φυσική μηχανή, δηλαδή τον υπολογιστή.

- **Notation (Σύνταξη και σημασία):** Οι δυσκολίες της ομάδας αυτής αφορούν στην εκάστοτε γλώσσα προγραμματισμού και τους κανόνες, που τη διέπουν. Ουσιαστικά, περιλαμβάνει τα προβλήματα, που προκύπτουν κατά τον προγραμματισμό λόγω συντακτικών λαθών ή ακόμα και λόγω χρήσης λανθασμένων εντολών.
- **Structures (Δομές):** Συγκεντρώνει τις δυσκολίες εκμάθησης των δομών. Πιο συγκεκριμένα, οι εκπαιδευόμενοι, όταν δεν κατανοούν σε βάθος τις δομές, αδυνατούν να τις ανακαλέσουν και να τις χρησιμοποιήσουν, με αποτέλεσμα να συναντούν αδιέξοδα κατά τη δόμηση ενός προγράμματος ή ακόμη να αναλώνονται στη συγγραφή κώδικα, που ήδη υπάρχει.
- **Pragmatics (Πρακτική εφαρμογή):** Περικλείει όλες τις δυσκολίες, που σχετίζονται με την ικανότητα καθορισμού, ανάπτυξης, ελέγχου και αποσφαλμάτωσης του προγράμματος με χρήση των εκάστοτε διαθέσιμων εργαλείων. Ουσιαστικά, αναφέρεται στη δυσκολία των εκπαιδευόμενων να προσαρμόζονται στα διάφορα περιβάλλοντα ανάπτυξης προγραμμάτων, με αποτέλεσμα να μην μπορούν να εκμεταλλευτούν όλες τις δυνατότητες, που προσφέρουν.

Παράλληλα, η έρευνα οδήγησε στην κατηγοριοποίηση των βασικών λαθών των εκπαιδευόμενων (Du Boulay, 1989; Εφόπουλος, 2005):

- **Κακή εφαρμογή της αναλογίας:** Ένα μεγάλο ποσοστό εκπαιδευόμενων θεωρεί ότι μια μεταβλητή μπορεί να αποθηκεύσει ταυτόχρονα περισσότερες από μία τιμές.
- **Υπεργενίκευση:** Οι εκπαιδευόμενοι συχνά θεωρούν πως ό,τι εφαρμόζεται για έναν τύπο αντικειμένου ή δομή, εφαρμόζεται αυτόματα και για οποιαδήποτε άλλη δομή-τύπο αντικειμένου.
- **Λάθος χειρισμός της πολυπλοκότητας και της αλληλεπίδρασης:** Οι εκπαιδευόμενοι συχνά αδυνατούν να διαχειριστούν τα επιμέρους τμήματα ενός μεγάλου προγράμματος ή τα τοποθετούν σε λάθος σημεία.

Συνδυάζοντας τις ομάδες δυσκολιών και τα βασικά λάθη των αρχάριων προγραμματιστών, γίνεται σαφές ότι ο προγραμματισμός είναι μια πολύπλοκη διαδικασία. Είναι κατανοητό ότι οι ομάδες δυσκολιών περιλαμβάνουν τις περιοχές εκείνες, στις οποίες εντοπίζονται τα σφάλματα και οι παρανοήσεις κατά τον προγραμματισμό. Η κατανόηση και αφομοίωση όλων των εννοιών, που περιγράφουν, αλλά και η πρακτική εφαρμογή τους αποτελούν την κυριότερη πρόκληση για τον αρχάριο προγραμματιστή.

Είναι γενικά αποδεκτό ότι η διαμόρφωση των κατάλληλων εκπαιδευτικών προσεγγίσεων σε συνδυασμό με τον άρτια καταρτισμένο εκπαιδευτή αναμένεται να φέρει το επιθυμητό αποτέλεσμα και να επιτύχει τον στόχο της εκπαιδευτικής διαδικασίας. Αντίστοιχα, κατά τη διδασκαλία του προγραμματισμού ένας ανάλογος συνδυασμός δύναται να οδηγήσει τον εκπαιδευόμενο στην αποτελεσματική αντιμετώπιση των δυσκολιών, παρανοήσεων και λαθών μέσα από την αποσαφήνιση και κατανόηση προγραμματιστικών εννοιών.

Παράλληλα, ιδιαίτερα καθοριστικό ρόλο στην εκπαιδευτική διαδικασία κατέχει το προγραμματιστικό περιβάλλον, που θα χρησιμοποιηθεί. Τα περιβάλλοντα, που επιλέγονται για τη διδασκαλία σε αρχάριους προγραμματιστές, χρησιμοποιούν ελκυστικά προς τον χρήστη γραφικά και προάγουν έναν απλοποιημένο τρόπο προσέγγισης μεταξύ εφαρμογής και χρήστη/εκπαιδευόμενου. Η χρήση οπτικών στοιχείων συμβάλλει σημαντικά τόσο στην κατανόηση, όσο και στην εφαρμογή βασικών αλγοριθμικών δομών, δίνοντας στη συνέχεια τη δυνατότητα στον εκπαιδευόμενο να «κατακτήσει» και τις παραδοσιακές γλώσσες προγραμματισμού.

Γενικότερα, η ανάγκη αντιμετώπισης των δυσκολιών και σφαλμάτων των εκπαιδευόμενων κατά τον προγραμματισμό καθιστά απαραίτητη την εύρεση νέων τεχνικών και προσεγγίσεων στην εκπαιδευτική διαδικασία, η οποία οφείλει να προσαρμόζεται και να εναρμονίζεται με τις απαιτήσεις της εποχής, να παρουσιάζει ένα προσιτό και σύγχρονο πρόσωπο, ώστε να ελκύει το ενδιαφέρον των εκπαιδευόμενων.

1.4. Προγραμματισμός και παιχνίδια

Είναι γενικά αποδεκτό ότι ο προγραμματισμός είναι μία σύνθετη και πολύπλευρη διαδικασία. Απαιτεί την κατανόηση βασικών προγραμματιστικών εννοιών, ώστε να είναι εφικτή η εφαρμογή της αποκτηθείσας γνώσης στην επίλυση προβλημάτων. Οι δυσκολίες, όμως, που προκύπτουν κατά τη διαδικασία του προγραμματισμού, συχνά αποθαρρύνουν τον αρχάριο προγραμματιστή. Κατά συνέπεια, προέκυψε ως επιτακτική ανάγκη η αναζήτηση νέων τεχνικών και προσεγγίσεων, προκειμένου η εκπαιδευτική διαδικασία να επιτύχει τον στόχο της, δηλαδή να προσφέρει τα κίνητρα εκείνα, που θα διατηρήσουν αμείωτο το ενδιαφέρον των εκπαιδευόμενων (Prensky, 2003), ώστε να επιτευχθεί η μάθηση.

Στην προσπάθεια αυτή, η προσοχή των ερευνών στράφηκε στα ενδιαφέροντα των εκπαιδευόμενων και ανέδειξε τη δυνατότητα χρήσης των ψηφιακών παιχνιδιών στην εκπαιδευτική διαδικασία. Ολοένα και περισσότεροι ερευνητές (Dempsey et al., 2002; Rosas et al., 2003; Pivec et al., 2004; Connolly et al., 2008 κ.α.) υποστηρίζουν ότι τα ψηφιακά παιχνίδια αποτελούν έναν ελκυστικό τρόπο προσέγγισης των νέων, μιας και είναι ιδιαίτερα δημοφιλή σε αυτούς, και αν

χρησιμοποιηθούν σωστά κατά τη διδασκαλία, είναι πολύ πιθανό να οδηγήσουν στην επίτευξη των μαθησιακών στόχων.

1.4.1. Ψηφιακά παιχνίδια

Τόσο στην ελληνική, όσο και στην ξένη βιβλιογραφία, απαντάται ένα ευρύ φάσμα όρων (Πανουτσόπουλος, 2010) για την απόδοση της έννοιας των ψηφιακών παιχνιδιών. Οι όροι αυτοί σχετίζονται μεταξύ τους και συχνά ταυτίζονται, παρόλο που παρουσιάζουν ορισμένες μικρές διαφορές. Ενδεικτικά, αναφέρουμε ορισμένους από αυτούς, όπως **βιντεοπαιχνίδια** (video games), **ηλεκτρονικά παιχνίδια** (electronic games), **παιχνίδια υπολογιστών** (computer games), **ψηφιακά παιχνίδια** (digital games) κ.α.

Στην παρούσα εργασία, θα χρησιμοποιηθεί ο όρος «ψηφιακό παιχνίδι», καθώς οι υπόλοιποι όροι αφορούν κατά κύριο λόγο το μέσο - συσκευή, που χρησιμοποιείται για την εκτέλεση του παιχνιδιού. Ο όρος αυτός δείχνει να είναι πιο περιεκτικός και χαρακτηρίζει ακριβέστερα όλες εκείνες τις ψηφιακές εφαρμογές, που στόχο έχουν να καταστήσουν ελκυστική τη διαδικασία της μάθησης και να επιτύχουν εκπαιδευτικούς και μαθησιακούς στόχους.

1.4.1.1. Ορισμός

Η έννοια του ψηφιακού παιχνιδιού δεν είναι δυνατόν να αποδοθεί από έναν και μόνο ορισμό, καθώς πρόκειται για ευρεία έννοια με πολλές παραμέτρους. Κατά καιρούς, αρκετοί ερευνητές έχουν διατυπώσει ορισμούς για το ψηφιακό παιχνίδι, εστιάζοντας κυρίως στα χαρακτηριστικά, που το προσδιορίζουν.

Σύμφωνα με τη Μαυρομάτη (2010), ο όρος ψηφιακό παιχνίδι αφορά σε μια ευρεία ποικιλία ψηφιακών εφαρμογών, τις οποίες χαρακτηρίζουν το παιγνιώδες περιβάλλον, η έντονη συμμετοχή του παίκτη και η χρήση πολυμεσικών στοιχείων.

Ο Κόμης (2004) αναφέρει ότι τα ψηφιακά παιχνίδια συνιστούν εκπαιδευτικό λογισμικό, το οποίο, χρησιμοποιώντας τη συνισταμένη παιχνίδι - κατά κανόνα θετική έννοια -, οδηγεί ευκολότερα στην υλοποίηση των μαθησιακών στόχων.

Πιο αναλυτικά, οι Kirriemuir J. & McFarlane (2004), που θεωρούν ότι τα ψηφιακά παιχνίδια διατηρούν σημαντικό ρόλο στην ψυχαγωγία των νέων σε καθημερινή βάση, ορίζουν ως ψηφιακό το παιχνίδι, που πληροί τις ακόλουθες προϋποθέσεις:

- Παρέχει οπτική πληροφορία σε έναν ή περισσότερους παίκτες

- Δίνει τη δυνατότητα στους παίκτες/χρήστες να εισάγουν δεδομένα
- Διέπεται από συγκεκριμένους κανόνες, με βάση τους οποίους διαχειρίζεται τα δεδομένα
- Τροποποιεί τις ψηφιακές πληροφορίες στους παίκτες
- Παίζεται σε κονσόλες, υπολογιστές και φορητές συσκευές

Σε μία γενικότερη περιγραφή ο Prensky (2007), που κατατάσσει τα ψηφιακά παιχνίδια στην κατηγορία των ενεργητικών εκπαιδευτικών περιβάλλοντων, υποστηρίζει ότι ο συνδυασμός παιχνιδιού και υπολογιστή στα ψηφιακά παιχνίδια παρέχει σε αυτά ελκυστικά χαρακτηριστικά, όπως γραφικά, κίνηση, μουσική κ.α., με αποτέλεσμα να είναι ενεργή και ευχάριστη η συμμετοχή του εκπαιδευόμενου.

Συνοψίζοντας, θα λέγαμε ότι το ψηφιακό παιχνίδι περιλαμβάνει απαραίτητα τη χρήση ηλεκτρονικής συσκευής (π.χ. κονσόλα, ηλεκτρονικός υπολογιστής (PC, laptop), κινητό τηλέφωνο, tablet κ.α.) - (με οθόνη ή συνδεδεμένη με οθόνη), με την οποία αλληλεπιδρά ο χρήστης/παίκτης, απαρτίζεται από κανόνες και στόχους, δύναται να παρέχει διάδραση μεταξύ περισσότερων χρηστών/παικτών, παρέχει ευχάριστο και ελκυστικό περιβάλλον, ενώ παράλληλα συμβάλλει στην απόκτηση γνώσεων και δεξιοτήτων.

1.4.1.2. Χαρακτηριστικά

Σύμφωνα με τις Νικηφορίδου & Παγγέ (2011), το ψηφιακό παιχνίδι είναι ένα σύγχρονο εργαλείο ικανό να ενισχύσει την εκπαιδευτική διαδικασία, με την προϋπόθεση ότι θα χρησιμοποιηθεί σωστά, δηλαδή θα εντάσσεται σε ένα οργανωμένο πλαίσιο εκπαίδευσης, με μαθησιακούς στόχους και κατάλληλο σχεδιασμό.

Η παραπάνω προϋπόθεση σε συνδυασμό με την πληθώρα ψηφιακών παιχνιδιών έκρινε αναγκαία τη διατύπωση των βασικών χαρακτηριστικών τους, ώστε να γίνεται η επιλογή του κατάλληλου κάθε φορά ψηφιακού παιχνιδιού. Ο εκπαιδευτικός οφείλει να είναι σε θέση να εντοπίσει τα χαρακτηριστικά αυτά, να αξιολογήσει το παιχνίδι και να επιλέξει εκείνο, που ταιριάζει περισσότερο στο αντικείμενο, που προτίθεται να διδάξει.

Τα βασικά χαρακτηριστικά των ψηφιακών παιχνιδιών, τα οποία φέρουν πολλά κοινά χαρακτηριστικά με το παιχνίδι (Prensky, 2001) και διατυπώθηκαν από τον Cudworth (1996), είναι τα εξής:

- Μαθησιακοί στόχοι
- Σύνολο κανόνων
- Αλληλεπίδραση

- Ανατροφοδότηση
- Ανταγωνισμός
- Στοιχείο πρόκλησης
- Στοιχείο διασκέδασης και κινήτρου
- Προϋπάρχουσα γνώση

Αναλύοντας τα χαρακτηριστικά αυτά, είναι αντιληπτό ότι ένα ψηφιακό παιχνίδι δημιουργείται για κάποιο σκοπό, οδηγεί στην επίτευξη ενός στόχου και διέπεται από ένα σύνολο κανόνων μέσα σε ένα δομημένο περιβάλλον. Ο χρήστης/παίκτης βρίσκεται σε αλληλεπίδραση με το παιχνίδι, έχει ενεργή συμμετοχή και απασχόληση και οι αποφάσεις του καθορίζουν την επίτευξη του στόχου. Από το ψηφιακό παιχνίδι δεν είναι δυνατόν να λείπει η έννοια της ανατροφοδότησης, δηλαδή η επιβράβευση και η «τιμωρία» ανάλογα με τις ενέργειες του χρήστη/παίκτη. Η ανατροφοδότηση, δείχνοντας τις θετικές και αρνητικές κινήσεις, συμβάλλει δραστικά στη βελτίωση της απόδοσης. Εξίσου σημαντικό ρόλο διαδραματίζει και ο ανταγωνισμός, που δύναται να οδηγήσει γρηγορότερα στον στόχο του παιχνιδιού και να ενισχύσει την αυτοπεποίθηση του χρήστη/παίκτη, ενώ το στοιχείο της πρόκλησης (βαθμός δυσκολίας, επίπεδα, γρίφοι) θα πρέπει να είναι προσαρμοσμένο στις δυνατότητες των εκπαιδευόμενων (Loftus and Loftus, 1983), προκειμένου να είναι εφικτή η ολοκλήρωση του παιχνιδιού, αλλά και να ενισχύεται η δημιουργικότητα. Τέλος, το ψηφιακό παιχνίδι έχει ως προϋπόθεση κάποια γνώση επάνω σε ένα αντικείμενο και παράλληλα, πρέπει να είναι ελκυστικό και διασκεδαστικό, ώστε να διατηρεί αμείωτο το ενδιαφέρον των εκπαιδευόμενων, προκειμένου να επιτευχθούν οι εκπαιδευτικοί στόχοι.

1.4.1.3. Θετικές και αρνητικές επιδράσεις

Πολλές μελέτες έχουν γίνει αναφορικά με τα πλεονεκτήματα και τα μειονεκτήματα, που δύναται να έχει η χρήση ψηφιακών παιχνιδιών στην εκπαιδευτική διαδικασία. Πρόκειται για ένα θέμα, που τίθεται ολοένα και περισσότερο από τους ερευνητές, καθώς η εξέλιξη της τεχνολογίας και η αλλαγή των ενδιαφέροντων των εκπαιδευόμενων, τείνουν να καταστήσουν ανιαρές τις παραδοσιακές μορφές διδασκαλίας.

Η αξιοποίηση των ψηφιακών παιχνιδιών στη διαδικασία της μάθησης φαίνεται να κερδίζει ολοένα και περισσότερους υποστηρικτές (Prensky, 2007; Gee, 2003; Βοσνιάδου, 2006; Μαλλιαράκης, Ξυνόγαλος & Σατρατζέμη, 2012 κ.α). Ειδικότερα, όσον αφορά στη διδασκαλία του προγραμματισμού φαίνεται να ξεπερνώνται αρκετά προβλήματα, που σχετίζονται με τη δυσκολία κατανόησης των εννοιών προγραμματισμού και τη χρήση κώδικα, αφού ο εκπαιδευόμενος διδάσκεται

μέσα από τη σχεδίαση και ανάπτυξη ενός παιχνιδιού. Η ίδια η έννοια παιχνίδι είναι θετική για εκείνον, ενώ του δίνεται η δυνατότητα να κατανοήσει τις βασικές αρχές του προγραμματισμού, χωρίς να προβληματιστεί με τη συγγραφή κώδικα, που απαιτεί την πρότερη γνώση κάποιας γλώσσας προγραμματισμού.

Η δημιουργία ενός ψηφιακού παιχνιδιού δύναται να λειτουργήσει θετικά στην εκμάθηση του προγραμματισμού, καθώς οι εκπαιδευόμενοι ασχολούνται με τα παιχνίδια αυτά στην καθημερινότητά τους (Cheng, M., Chen, J., Chu, S. et al., 2015). Άλλωστε, πρόκειται για μία ελκυστική και διασκεδαστική ενασχόληση, κατά την οποία ο εκπαιδευόμενος, στο πλαίσιο ενός σχεδίου μαθήματος, διαμορφώνει την εξέλιξη του παιχνιδιού με τις δικές του αποφάσεις και ενέργειες, μέσα από τον πειραματισμό, ακολουθώντας έναν δικό του ρυθμό μάθησης (Μαλλιαράκης, Ξυνόγαλος & Σατρατζέμη, 2012). Αναπτύσσει μεθόδους και στρατηγικές, προκειμένου να ολοκληρώσει με επιτυχία το παιχνίδι, ενώ σε όλη τη διάρκεια λαμβάνει άμεση ανατροφοδότηση (Gee, 2003, Βοσνιάδου, 2006), ώστε να μπορεί να διαχειριστεί τυχόν λάθη ή/και να απολαύσει την επιτυχία. Με λίγα λόγια, αποτελεί μέρος της εκπαιδευτικής διαδικασίας, συμμετέχει ενεργά σε αυτή και αποκτά εμπειρίες, τις οποίες εύκολα θα αξιοποιήσει στο μέλλον, επιτυγχάνοντας τη μεταγνώση (Papert, 1999).

Σύμφωνα με τον Prensky (2003), το ψηφιακό παιχνίδι, που είναι μέρος της καθημερινότητας των εκπαιδευόμενων, συμβάλλει ενεργά και αποτελεσματικά στη μάθηση, αφού επιτυγχάνει να τα υποκινήσει να συμμετάσχουν σε αυτή, σε αντίθεση με τις παραδοσιακές μεθόδους εκπαίδευσης. Αυτός ο τρόπος διδασκαλίας απαιτεί τη διάδραση, μια διαδικασία που προσαρμόζει τους μαθησιακούς στόχους ανάλογα με τις εκάστοτε ανάγκες. Λόγω των χαρακτηριστικών των ψηφιακών παιχνιδιών και της αποδοχής, που αυτά λαμβάνουν, θεωρούνται αναπόσπαστο μέρος της εκπαιδευτικής διαδικασίας.

Ωστόσο, σε αυτό το σημείο κρίνεται απαραίτητο να αναφερθούν ορισμένα μειονεκτήματα (αρνητικά στοιχεία), που σχετίζονται με τη χρήση των ψηφιακών παιχνιδιών κατά τη μάθηση.

Ξεκινώντας από τα λειτουργικά θέματα, οφείλουμε να αναφέρουμε την έλλειψη εξοπλισμού σε ορισμένες σχολικές μονάδες, την έλλειψη εξειδίκευσης στον προγραμματισμό ορισμένων εκπαιδευτικών και την έλλειψη χρόνου. Είναι γεγονός ότι κάποιες σχολικές μονάδες δεν έχουν εργαστήριο πληροφορικής, ενώ άλλες αδυνατούν να διαθέσουν έναν υπολογιστή σε κάθε εκπαιδευόμενο, με αποτέλεσμα να μοιράζονται τον ίδιο υπολογιστή δύο ή και περισσότεροι εκπαιδευόμενοι. Επίσης, η χρήση ψηφιακών παιχνιδιών στην εκπαίδευση δεν έχει ενταχθεί οργανωμένα στο αναλυτικό πρόγραμμα, με αποτέλεσμα η χρήση τους να έγκειται στη διάθεση του εκάστοτε εκπαιδευτικού. Άλλωστε, αυτός ο τρόπος διδασκαλίας δεν μπορεί να ολοκληρωθεί σε λίγες ώρες διδασκαλίας. Απαιτεί πολύ χρόνο, ώστε κάθε εκπαιδευόμενος να κινηθεί με τον δικό του ρυθμό, πάντα βέβαια στο πλαίσιο, που έχει ορίσει ο εκπαιδευτικός.

Περνώντας στις αρνητικές επιδράσεις των παιχνιδιών στους εκπαιδευόμενους, είναι απαραίτητο να αναφερθεί ότι οι περισσότεροι ερευνητές, που υποστηρίζουν τη χρήση ψηφιακών παιχνιδιών στην εκπαιδευτική διαδικασία, αποδέχονται και την ύπαρξη κάποιων αρνητικών συνεπειών. Θεωρείται, λοιπόν, ότι τα παιχνίδια αυτού του είδους ενισχύουν την επιθετικότητα (Anderson, 2003). Είναι βέβαιο ότι τα ψηφιακά παιχνίδια στην πλειοψηφία τους προβάλλουν σκηνές βίας, γεγονός ιδιαίτερα αρνητικό ως προς την επίδρασή του στον χρήστη. (Αξίζει στο σημείο αυτό να διευκρινιστεί ότι τα εκπαιδευτικά ψηφιακά παιχνίδια δεν είναι παιχνίδια βίας και όσα καλούνται να δημιουργήσουν οι εκπαιδευόμενοι στη σχολική τάξη, υπό την καθοδήγηση του εκπαιδευτικού, δεν έχουν τέτοια χαρακτηριστικά).

Μία ακόμη αρνητική συνέπεια είναι η απομόνωση. Παρόλο που ορισμένα διαδικτυακά παιχνίδια παίζονται με πολλούς παίκτες, είναι γεγονός ότι ο χρήστης περνά αρκετές ώρες μόνος του, χάνει την επαφή με τους συνομήλικούς του και δεν κοινωνικοποιείται. Όμως, και σε αυτήν την περίπτωση, καθώς οι εκπαιδευόμενοι βρίσκονται στη σχολική τάξη είναι βέβαιο ότι θα υπάρχει κάποια συνεργασία υπό την καθοδήγηση και του εκπαιδευτικού.

Τέλος, ένα ακόμη θέμα σχετικά με τα ψηφιακά παιχνίδια, που απασχολεί ιδιαίτερα και τους γονείς, είναι ο εθισμός (Grusser, Thalemann and Griffiths, 2007). Ανεξάρτητα από την ηλικία του εκπαιδευόμενου, η πολύωρη ενασχόληση μπορεί να οδηγήσει το άτομο στον εθισμό, με αποτέλεσμα να εξαρτάται από το παιχνίδι και να μην μπορεί να απομακρυνθεί από αυτό.

Ολοκληρώνοντας, είναι σημαντικό να επισημανθεί ότι όλα τα ψηφιακά παιχνίδια δεν είναι ίδια. Υπάρχουν παιχνίδια, τα οποία μπορούν να χρησιμοποιηθούν στην εκπαιδευτική διαδικασία με εξαιρετικά αποτελέσματα ως προς τη μάθηση και την επίτευξη των μαθησιακών στόχων. Έγκειται στην ευχέρεια του εκπαιδευτικού να βρει τα παιχνίδια εκείνα, που θα εξυπηρετήσουν όσο το δυνατόν καλύτερα τις ανάγκες του μαθήματος, θα διατηρήσουν αμείωτο το ενδιαφέρον των εκπαιδευόμενων, καθιστώντας τους μέρος της μαθησιακής διαδικασίας, και θα οδηγήσουν στην απόκτηση της γνώσης. Ειδικά, για τη διδασκαλία του προγραμματισμού η δυνατότητα δημιουργίας (σχεδίασης και ανάπτυξης) ενός παιχνιδιού εξ ολοκλήρου από τον εκπαιδευόμενο οδηγεί τόσο στην επίτευξη των μαθησιακών στόχων, όσο και στην απομάκρυνση όλων των αρνητικών επιδράσεων, που αναφέρθηκαν παραπάνω.

1.4.1.4. Ένταξη στην εκπαιδευτική διαδικασία

Η επιλογή της διδασκαλίας με χρήση παιχνιδιών καλεί τον εκπαιδευτικό να επιλέξει τον τρόπο με τον οποίο θα εμπλακούν οι εκπαιδευόμενοι στη διαδικασία της μάθησης. Μετά την ολοκλήρωση του θεωρητικού πλαισίου του μαθήματος απαιτείται η χρήση της τεχνικής εκείνης, που θα οδηγήσει ομαλά και αποτελεσματικά στην επίτευξη των μαθησιακών στόχων. Στην περίπτωση της

παιχνιδοκεντρικής μάθησης και ειδικότερα στη διδασκαλία του προγραμματισμού απαντώνται τρεις τεχνικές ανάλογα με το λογισμικό, που θα χρησιμοποιηθεί.

Η πρώτη τεχνική αφορά στη χρήση ενός έτοιμου παιχνιδιού. Σύμφωνα με τους Vos, N., van der Meijden, H. & Denessen, E. (2011), οι εκπαιδευόμενοι αποκομίζουν περισσότερα οφέλη, όταν σχεδιάζουν το παιχνίδι, παρά όταν απλά παίζουν. Οι εκπαιδευόμενοι παίζουν ένα υπάρχον παιχνίδι, χωρίς να χρειαστεί να προγραμματίσουν, και λαμβάνουν ανατροφοδότηση με βάση τις κινήσεις τους σε αυτό και τους κανόνες που το διέπουν. Η δυσκολία στην τεχνική αυτή έγκειται στο γεγονός ότι δεν είναι εύκολο ένα παιχνίδι να ταιριάζει απόλυτα στο γνωστικό αντικείμενο, ώστε να καλύπτει και τους μαθησιακούς στόχους, που έχουν τεθεί από τον εκπαιδευτικό.

Στη δεύτερη τεχνική, η οποία χρησιμοποιήθηκε στην παρούσα εργασία για τη διδασκαλία του Προγραμματισμού, αξιοποιώντας τα ψηφιακά παιχνίδια, δίνεται στους εκπαιδευόμενους ένα παιχνίδι ημιτελές (μισο-έτοιμο, όπως συνηθίζεται να λέγεται), με το οποίο καλούνται να πειραματιστούν. Ο στόχος είναι να εργαστούν οι εκπαιδευόμενοι με το υπάρχον προσχέδιο, ώστε χρησιμοποιώντας τις γνώσεις, που αποκόμισαν κατά το θεωρητικό πλαίσιο του μαθήματος, να το επεκτείνουν. Σημαντικό πλεονέκτημα αυτής της τεχνικής είναι ο χρόνος, που κερδίζεται, αποφεύγοντας πολύπλοκες προγραμματιστικές έννοιες και όρους, που σε αυτό το στάδιο θα μπερδέψουν και ενδεχομένως θα αποθαρρύνουν τον εκπαιδευόμενο.

Τέλος, η τρίτη περίπτωση αφορά στην ανάπτυξη ενός παιχνιδιού εξ ολοκλήρου από τους εκπαιδευόμενους, γεγονός που προσφέρει ευελιξία στον εκπαιδευτικό κατά τον ορισμό των μαθησιακών στόχων. Η τεχνική αυτή ενισχύει τη δημιουργικότητα των εκπαιδευόμενων, αλλά προϋποθέτει να έχουν κάποιες βασικές γνώσεις στο αντικείμενο, ειδάλλως θα απαιτηθούν πολλές ώρες διδασκαλίας, προκειμένου να είναι σε θέση να χρησιμοποιήσουν ένα προγραμματιστικό περιβάλλον. Έτσι, υπάρχει πάντα ο κίνδυνος να κουραστούν και μειωθεί το ενδιαφέρον τους. Παράλληλα, ο εκπαιδευτικός θα πρέπει να είναι σε θέση να διδάξει αναλυτικά το προγραμματιστικό περιβάλλον και να παρέχει ικανή υποστήριξη, ώστε να μην αποθαρρυνθούν οι εκπαιδευόμενοι κατά τα πρώτα προγραμματιστικά βήματά τους.

1.4.2. Εκπαιδευτικά περιβάλλοντα – Σύγκριση

Τα τελευταία χρόνια έχουν αναπτυχθεί πολλά και διαφορετικά εκπαιδευτικά ψηφιακά περιβάλλοντα, προκειμένου να διδαχθούν οι βασικές αρχές και έννοιες του προγραμματισμού με ένα πιο προσιτό και κατανοητό τρόπο. Τα περιβάλλοντα αυτά μπορούν να χρησιμοποιηθούν κατά την εκπαιδευτική διαδικασία με τη δημιουργία παιχνιδιών από τους ίδιους τους εκπαιδευόμενους.

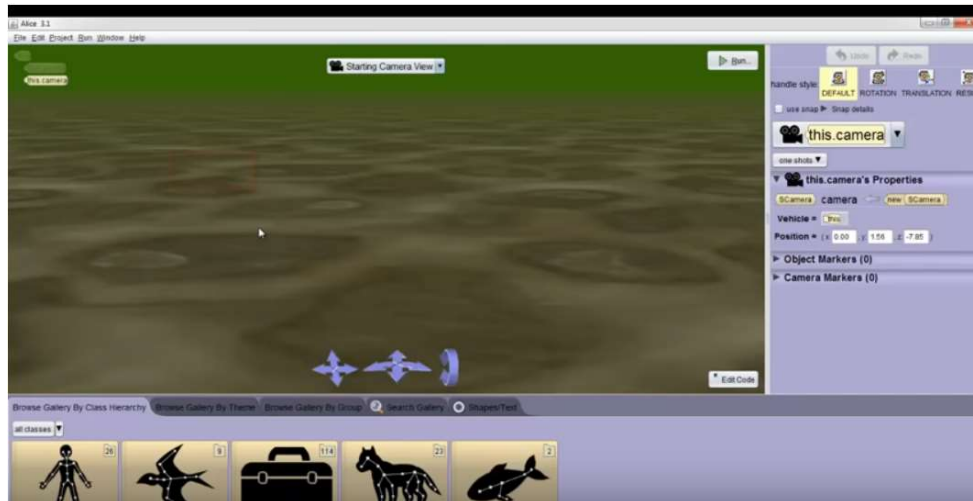
Κάθε ένα από αυτά φέρει συγκεκριμένα χαρακτηριστικά, δυνατότητες, αλλά και περιορισμούς. Ο εκπαιδευτικός καλείται να επιλέξει το εργαλείο εκείνο, που εξυπηρετεί καλύτερα τη μαθησιακή διαδικασία και είναι δυνατόν να επιτύχει τους στόχους, που έχουν τεθεί.

Για την παρούσα εργασία, έγινε σύγκριση μεταξύ τριών από αυτά τα συγγραφικά εργαλεία ανάπτυξης, ώστε να επιλεγεί το κατάλληλο, εκείνο δηλαδή το οποίο θα πληροί τις προϋποθέσεις και τους όρους, που διέπουν την παρούσα εργασία.

Με βάση λοιπόν τα δεδομένα της εργασίας, το εκπαιδευτικό περιβάλλον θα έπρεπε να απευθύνεται σε ενήλικες εκπαιδευόμενους, που έχουν ήδη βασικές γνώσεις στον προγραμματισμό C, σύμφωνα με τον οδηγό προγράμματος σπουδών του τμήματος, να παρέχει τη δυνατότητα εκμάθησης της γλώσσας προγραμματισμού Java σε εισαγωγικό βαθμό και ανάλογα με τις μαθησιακές ανάγκες των εκπαιδευόμενων. Παράλληλα, θα έπρεπε αν είναι μία ελεύθερη, μη συνδρομητική πλατφόρμα σχεδίασης και ανάπτυξης, που θα περιλαμβάνει εγκατεστημένη βάση υποστήριξης από έναν αρκετά μεγάλο αριθμό χρηστών χωρίς τη χρήση ειδικών αδειών, ώστε οι εκπαιδευόμενοι να έχουν τη δυνατότητα να εργαστούν ακόμη και από το σπίτι, αν αυτό θεωρηθεί απαραίτητο. Τέλος, οι εκπαιδευόμενοι θα έπρεπε να έχουν πλήρη διαχειριστικό και εποπτικό έλεγχο της σε όλες τις φάσεις ανάπτυξης.

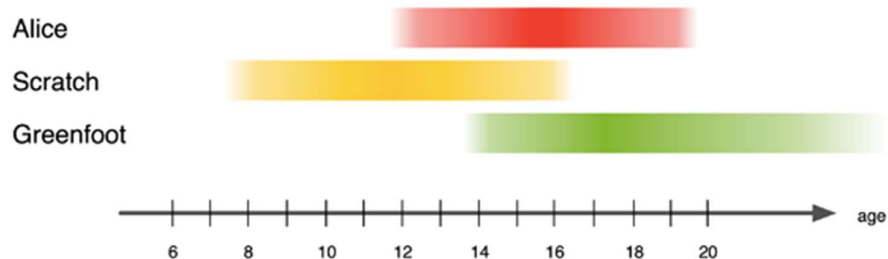
Μετά από διερεύνηση ορισμένων εργαλείων (Kodu, GameMaker, Sploder, Scratch, Alice, Greenfoot κ.λπ) επιλέχθηκαν προς σύγκριση τρία προγραμματιστικά περιβάλλοντα: το **Alice**, το **Scratch** και το **Greenfoot**, ώστε να ελεγχθούν οι δυνατότητές τους και να χρησιμοποιηθεί τελικά εκείνο, που θα καλύπτει τις προϋποθέσεις, που τέθηκαν παραπάνω. Ιδιαίτερα βοηθητική σε αυτή την προσπάθεια ήταν η έρευνα πέντε συγγραφέων (Utting, I., Cooper, S., Kölling, M., Maloney, J. & Resnick, M., 2010), οι οποίοι συνέκριναν τα τρία αυτά εργαλεία ως προς τις δυνατότητες και τους περιορισμούς τους. Σε αυτό το σημείο, ας επισημανθεί ότι και τα τρία εργαλεία παρέχονται δωρεάν, δεν απαιτείται κάποια συνδρομή κατά τη χρήση τους και έχουν τη δυνατότητα στήριξης μεγάλου αριθμού χρηστών χωρίς ειδικές άδειες.

Ξεκινώντας από το εκπαιδευτικό ψηφιακό περιβάλλον **Alice**, διαπιστώνει κανείς ότι αποτελεί μία ενδιαφέρουσα πλατφόρμα, που απευθύνεται σε μεγαλύτερης ηλικίας εκπαιδευόμενους και χρησιμοποιεί τη γλώσσα Java. Ωστόσο, το δύσκολο περιβάλλον διεπαφής σε συνδυασμό με τις έννοιες της τρισδιάστατης απεικόνισης, όπως προβολή κάμερας, εστίαση, φωτισμοί κτλ, πιθανότατα να δυσκολέψει τους εκπαιδευόμενους και να λειτουργήσει αποθαρρυντικά για αυτούς.



Εικόνα 1. Το προγραμματιστικό περιβάλλον Alice

Το εργαλείο σχεδιασμού και ανάπτυξης **Scratch** είναι ιδιαίτερο φιλικό και κατανοητό στον χρήστη, που καθώς δημιουργεί ένα παιχνίδι, διδάσκεται παράλληλα και βασικές αρχές του προγραμματισμού. Ωστόσο, δεν πληροί δύο βασικές προϋποθέσεις: α) εστιάζει περισσότερο στη διαχείριση των μέσων (Comparing Alice, Greenfoot & Scratch, SIGCSE 2010, σελ. 192-193), δίνοντας πρόσβαση σε προγραμματιστικές έννοιες με τη χρήση των blocks, κάτι που δεν ταιριάζει στις συγκεκριμένες μαθησιακές ανάγκες λόγω του ότι οι εκπαιδευόμενοι έχουν ήδη διδαχθεί τις βασικές έννοιες του προγραμματισμού και στοχεύουν περισσότερο στην εκμάθηση μιας νέας γλώσσας και β) πρόκειται για ένα λογισμικό, που φαίνεται να ταιριάζει περισσότερο σε παιδιά μικρότερης ηλικίας, αφού θεωρείται κυρίως εισαγωγικό εργαλείο.



Εικόνα 2. Οι ηλικιακές ομάδες, στις οποίες απευθύνονται τα περιβάλλοντα Alice, Scratch και Greenfoot (Woei, L.S., IH Othman I. H. & Man C. K. (2013))

Η πλατφόρμα **Greenfoot** διαθέτει φιλικό και προσιτό περιβάλλον διεπαφής (user interface). Σχεδιάστηκε ειδικά για την εκμάθηση της γλώσσας προγραμματισμού Java και απευθύνεται σε

χρήστες μεγαλύτερων ηλικιών (14 και άνω). Φαίνεται, λοιπόν, να ικανοποιεί όλες τις προϋποθέσεις, που τέθηκαν προηγουμένως.

Ενισχυτικά στην απόφαση να χρησιμοποιηθεί η πλατφόρμα Greenfoot στην παρούσα εργασία λειτούργησε ο πίνακας, που ακολουθεί (Τατόγλου Χ., 2018) και παρουσιάζει μελέτες, που διεξήχθησαν με στόχο την αποτελεσματικότητα του Greenfoot στην εκπαιδευτική διαδικασία.

ΣΥΝΟΠΤΙΚΟΣ ΠΙΝΑΚΑΣ ΜΕΛΕΤΩΝ		
Αναφορά	Στόχος μελέτης	Συμπέρασμα
Al-Bow et al. (2008)	Αξιολόγηση του Greenfoot ως προγραμματιστική πλατφόρμα, μέσα από πειραματική μελέτη	Τα αποτελέσματα του πειράματος, που διενεργήθηκε, έδειξαν ότι οι εκπαιδευόμενοι, οι οποίοι έλαβαν μέρος σε αυτό, ανέπτυξαν ικανότητες και δεξιότητες, ενώ ο τρόπος σκέψης τους διαφοροποιήθηκε σε σχέση με προηγουμένως.
Gallant & Mahmoud (2008)	Παρουσίαση της διδασκαλίας των βασικών αρχών του προγραμματισμού σε εκπαιδευόμενους μέσω του εκπαιδευτικού εργαλείου Greenfoot.	Στους εκπαιδευόμενους άρεσε το μάθημα στο εργαστήριο με τη χρήση του Greenfoot, το οποίο τους βοήθησε να κατανοήσουν τις έννοιες του προγραμματισμού, διασκεδάζοντας. Παράλληλα σημειώθηκε αύξηση στο ενδιαφέρον των εκπαιδευόμενων για τον προγραμματισμό και κυρίως σε στοιχεία τα οποία σχετίζονται με την αλληλεπίδραση του χρήστη με την εφαρμογή.
Rick et al. (2010)	Διεξαγωγή πειράματος με στόχο την αξιολόγηση του Greenfoot ως εργαλείου εκπαίδευσης των χρηστών στον προγραμματισμό.	Το Greenfoot ενδείκνυται για την εκτέλεση των εισαγωγικών μαθημάτων επιστήμης των υπολογιστών στην εκπαίδευση για την εκμάθηση των αρχικών εννοιών της επιχειρηματικής πληροφορικής.

Simmons, DiSalvo & Guzdial (2012)	Παρουσίαση της αξιοποίησης του Greenfoot για την εκμάθηση του προγραμματισμού μέσω της ανάπτυξης παιχνιδιών	Τα βασικά συμπεράσματα, που προέκυψαν από την παρατήρηση της συμπεριφοράς των εκπαιδευόμενων στην έρευνα αυτή, ήταν ότι, πρώτον, οι εκπαιδευόμενοι δεν είχαν ρεαλιστική αντίληψη των προγραμματιστικών εννοιών, που κατείχαν και δεύτερον, ότι καθώς οι συνεδρίες δυσκόλευαν οι εκπαιδευόμενοι συνειδητοποιούσαν ότι δεν είναι έτοιμοι ακόμα και όταν υπάρχουν έννοιες, τις οποίες γνωρίζουν.
Vilner, Zur & Tavor (2011)	Η έρευνα αυτή χρησιμοποιείται, για να αξιολογηθεί η κατανόηση εννοιών προγραμματισμού από εκπαιδευόμενους, όταν η εισαγωγή στον προγραμματισμό γίνεται μέσω της ανάπτυξης παιχνιδιών.	Οι εκπαιδευόμενοι ανέφεραν ότι το Greenfoot και το γραφικό περιβάλλον, που διαθέτει, τους βοήθησε να μάθουν εύκολα και διασκεδαστικά τόσο να προγραμματίζουν, όσο και να υλοποιούν παιχνίδια.
Jonas (2013)	Η έρευνα αυτή ασχολείται με τη χρήση επιτραπέζιων παιχνιδιών στρατηγικής πολλών παικτών για την εκμάθηση των βασικών αρχών του προγραμματισμού.	Μέσα από την παρουσίαση των αποτελεσμάτων σημειώνεται χαρακτηριστικά ότι οι εκπαιδευόμενοι έμαθαν τους κανόνες των παιχνιδιών στρατηγικής, τα οποία τους δόθηκαν, πρώτα μέσω της μηχανής παιχνιδιού του Greenfoot.
Hijon-Neira et al. (2014)	Η εν λόγω έρευνα παρουσιάζει το ProGames, ένα εκπαιδευτικό σύστημα,	Το συμπέρασμα της έρευνας αυτής παρουσιάζει ότι η χρήση του εργαλείου ProGames ως εκπαιδευτικού εργαλείου βελτιώνει τις γνώσεις των συμμετεχόντων αρκετά.

	που χρησιμοποιείται για τη διδασκαλία του προγραμματισμού σε εκπαιδευόμενους μέσω του εργαλείου Greenfoot. Το σύστημα αυτό φέρει ένα σύνολο από παιχνίδια, που είναι προγραμματιστικές ασκήσεις.	
Price et al. (2016)	Η έρευνα αυτή ασχολείται με την εμπειρική αξιολόγηση του επεξεργαστή πηγαίου κώδικα Stride του Greenfoot, συγκρίνοντας την απόδοση δύο ομάδων εκπαιδευόμενων κατά τη χρήση της πλατφόρμας.	Αναλύοντας συνολικά τα δεδομένα βγήκε το συμπέρασμα ότι ο επεξεργαστής πηγαίου κώδικα Stride είναι περισσότερο βοηθητικός, όσον αφορά στο χειρισμό των συντακτικών λαθών, γεγονός που τον καθιστά περισσότερο αποτελεσματικό.

Πίνακας 1. Συνοπτικός πίνακας μελετών σχετικά με το εργαλείο Greenfoot (Τατόγλου, 2018)

Το Greenfoot είναι ένα εργαλείο με αυξημένες δυνατότητες και αρκετά πλεονεκτήματα. Δίνει στον χρήστη τη δυνατότητα να απεικονίσει τα αντικείμενα ενός σεναρίου και να δημιουργήσει αλληλεπιδράσεις με αυτά. Ένα από τα σημαντικότερα χαρακτηριστικά, που το διαχωρίζει από τα υπόλοιπα περιβάλλοντα ανάπτυξης, είναι η άμεση αλληλεπίδραση, που μπορεί να έχει ο χρήστης με το περιβάλλον και γενικότερα με τον κόσμο, που δημιουργείται στο πλαίσιο του παιχνιδιού. Μέσω αυτής της αλληλεπίδρασης, ο χρήστης καθορίζει την εξέλιξη του παιχνιδιού με τις δικές του επιλογές και ενέργειες.

Το Greenfoot δημιουργήθηκε με σκοπό να αποτελέσει ένα γραφικό και αλληλεπιδραστικό περιβάλλον, ώστε να ελκύει περισσότερο σε σχέση με άλλα ανταγωνιστικά περιβάλλοντα. Η ανάπτυξη

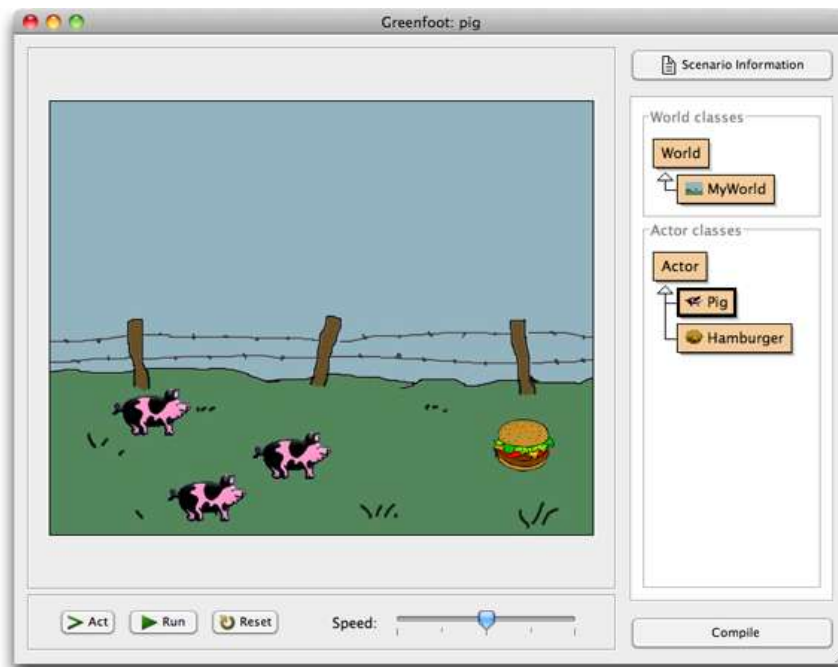
μιας ευρείας γκάμας διαφορετικών σεναρίων εφαρμογής, όπως προσομοιώσεις ή παιχνίδια, το καθιστούν μια ιδιαίτερα γρήγορη και εύκολη λύση. Το περιβάλλον εργασίας είναι ολοκληρωμένο (IDE) και περιλαμβάνει τη διαχείριση του έργου (project manager), την αυτόματη συμπλήρωση εντολών (auto completion), τη μορφοποίηση της σύνταξης (syntax highlighting) και πολλά άλλα εργαλεία, που συναντώνται σε επαγγελματικά προγραμματιστικά περιβάλλοντα. Παρόλα αυτά, σχεδιάστηκε με τέτοιο τρόπο, ώστε να είναι απλό, σαφές και κατανοητό, ειδικά για τον αρχάριο χρήστη.

Η γλώσσα προγραμματισμού, που χρησιμοποιείται στο Greenfoot, είναι η standard γλώσσα προγραμματισμού Java, που αποτελεί σήμερα την κυρίαρχη γλώσσα, τόσο στην ακαδημαϊκή κοινότητα, όσο και στη βιομηχανία. Η βασική σύλληψη της κεντρικής ιδέας για το πως λειτουργούν κάποια πράγματα, μεταφέρεται απευθείας στο εκπαιδευόμενο, ο οποίος στη συνέχεια προσαρμόζει την αποκτηθείσα γνώση αποτελεσματικά σε πιο πολύπλοκα, επαγγελματικά, προγραμματιστικά περιβάλλοντα εργασίας.

Τέλος, ο εκλεπτυσμένος και έξυπνος τρόπος, με τον οποίο κάποιος μπορεί να συνθέσει ακόμη και πιο περίπλοκα προβλήματα, κρατάει αμείωτο το ενδιαφέρον του εκπαιδευόμενου μέσω των οπτικών αποτελεσμάτων και φαίνεται να εκπληρώνει τις προσδοκίες του για την εισαγωγική επιμόρφωση στη συγκεκριμένη γλώσσα προγραμματισμού, δίνοντας συνάμα το έναυσμα να περάσει ομαλά στα γενικότερα προγραμματιστικά περιβάλλοντα της Java (IDE – Integrated Development Environments).

ΕΝΟΤΗΤΑ 2. ΕΚΠΑΙΔΕΥΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ GREENFOOT

2.1. Το προγραμματιστικό περιβάλλον Greenfoot



Εικόνα 3. Το περιβάλλον του Greenfoot

Το Greenfoot είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών (Integrated Development Environment - IDE), το οποίο χρησιμοποιείται για εκπαιδευτικούς σκοπούς και ειδικότερα για τη διδασκαλία του αντικειμενοστραφούς προγραμματισμού, με τη συνδρομή της γλώσσας προγραμματισμού Java ή Stride. Πρόκειται για ένα εκπαιδευτικό εργαλείο, που απευθύνεται τόσο σε αρχάριους, όσο και σε προχωρημένους χρήστες, καθώς δίνει τη δυνατότητα εμφάνισης των αποτελεσμάτων του κώδικα με οπτικό τρόπο. Επιτρέπει, δηλαδή, την εύκολη ανάπτυξη εφαρμογών δισδιάστατων γραφικών, όπως προσομοιώσεις (simulators) και διαδραστικά παιχνίδια.

Παράλληλα, το Greenfoot διευκολύνει την εκμάθηση του αντικειμενοστραφούς προγραμματισμού, καθώς μπορεί να επιτευχθεί η δημιουργία παιχνιδιών ακόμη και με γνώση μόνο των βασικών στοιχείων της Java.

Η εφαρμογή περιλαμβάνει μια ευρύτατη βιβλιοθήκη συναρτήσεων, η οποία καλύπτει οποιαδήποτε ενέργεια επιθυμεί ο χρήστης. Όλες οι συναρτήσεις της Java, αλλά και του Greenfoot, είναι διαθέσιμες στις οδηγίες (documentation) της εφαρμογής στην επιλογή Help, ώστε να μην είναι απαραίτητη η απομνημόνευσή τους και να προάγεται η εξάσκηση και εμπέδωση της αλγοριθμικής σκέψης. Το Greenfoot αναπτύσσεται και συντηρείται από το King's College London με την

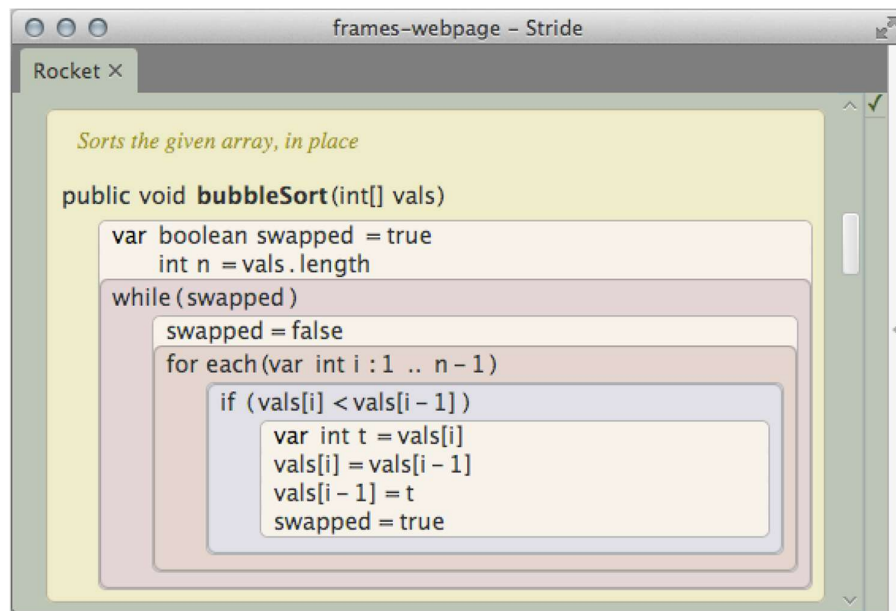
υποστήριξη της Oracle, δηλαδή της εταιρίας που δημιούργησε τη γλώσσα προγραμματισμού Java. Είναι δωρεάν, κάνοντας χρήση της άδειας GPL (GNU General Public License) και διατίθεται για λειτουργικά συστήματα Windows, macOS, Linux, Solaris και γενικά σε οποιαδήποτε σύγχρονη συσκευή διαθέτει Java Virtual Machine (JVM).

2.2. Ιστορία

Το Greenfoot ως project ξεκίνησε το 2003 από τον **Michael Kölling** (Γερμανός επιστήμονας της πληροφορικής, καθηγητής στο King's College του Λονδίνου) και το αρχικό πρωτότυπο φτιάχτηκε από τον μεταπτυχιακό φοιτητή Poul Henriksen, τον οποίο επέβλεπε ο Michael Kölling στα τέλη του 2003 και τις αρχές του 2004. Από το 2005 και μετά, η ανάπτυξη και εξέλιξη του έργου συνεχίστηκε με τη συνεργασία του BlueJ Group. Το Greenfoot σήμερα αναπτύσσεται και συντηρείται από το King's College London, όπου και εγκαταστάθηκε η ομάδα ανάπτυξης, η οποία μεταφέρθηκε από το πανεπιστήμιο του Kent. Η πρώτη έκδοση v1.00 δημοσιεύτηκε στις 31 Μαΐου 2006 και ακολούθησαν οι επόμενες σε μικρά χρονικά διαστήματα. Τον Μάιο του 2007, το έργο αυτό κέρδισε το βραβείο "Duke's Choice Award" στην κατηγορία "Java Technology in Education" και το 2010, το βραβείο "Premier Award for Excellence in Engineering Education Courseware". Τον Μάρτιο του 2009 το έργο Greenfoot έγινε δωρεάν και ανοιχτού κώδικα λογισμικό και αδειοδοτήθηκε με το GNU GPL και τον περιορισμό του Classpath exception, που αφορά εν ολίγοις τον τρόπο σύνδεσης (linking) των επιμέρους βιβλιοθηκών λογισμικού και τον τρόπο διανομής τους ως τελικό προϊόν από τους χρήστες, που θα τις χρησιμοποιήσουν.

Τον Αύγουστο του 2009 κυκλοφόρησε το πρώτο βιβλίο του Michael Kölling με τίτλο «**Introduction to Programming with Greenfoot**», που διδάσκει τον τρόπο προγραμματισμού του Greenfoot. Το 2017, το Greenfoot επεκτάθηκε, ώστε να υποστηρίζει και μια δεύτερη γλώσσα προγραμματισμού, εκτός της Java, που ονομάστηκε Stride. Η γλώσσα αυτή προστέθηκε, ώστε να κάνει τον προγραμματισμό ευκολότερο για τους αρχάριους προγραμματιστές, να υποστηρίζει περισσότερο τους νεότερους χρήστες και να προσθέσει ένα μεταβατικό στάδιο από τα block based περιβάλλοντα προγραμματισμού, π.χ. Scratch, στον κλασικό επεξεργαστή κειμένων και προγραμματισμό μέσω εντολών. Η κυριότερη διαφορά, που θα συναντήσει κανείς, είναι ο επεξεργαστής κειμένου (editor), ο οποίος είναι frame-based, δηλαδή εντολές ενθυλακωμένες μέσα σε πλαίσια.

Είναι άλλη μια καινοτομία του King's College London, που συνδυάζει τα πλεονεκτήματα των κλασικών επεξεργαστών κειμένου κώδικα και των block επεξεργαστών, όπως αυτό του Blockly και Scratch.



Εικόνα 4. Η αρχή και το τέλος των πλαισίων της Stride

Τα πλαίσια μέσα στον επεξεργαστή κειμένου αποτελούν κόμβους (nodes) μέσα σε ένα ιεραρχικό δέντρο εντολών και αποτελούνται από απλές εντολές, δομές όπως για παράδειγμα δομές επανάληψης, ελέγχου κτλ ή και μεθόδους, δηλαδή οι συναρτήσεις που ενθυλακώνονται μέσα σε μια κλάση. Χρησιμοποιούνται ως ενιαίες και αδιαίρετες οντότητες κατά τον χειρισμό τους μέσα από το περιβάλλον διεπαφής (user interface). Συνδυάζουν ταυτόχρονη χρήση κειμένου και γραφικών στοιχείων, ενώ η διαχείρισή τους γίνεται με κινήσεις του ποντικιού (mouse-based gestures) και του πληκτρολογίου. Ο χρωματικός κώδικας, που χρησιμοποιείται για την οπτικοποίησή τους, διαχωρίζει τους κόμβους-πλαίσια καθώς και τα σημεία των χειρισμών τους.

Το πλεονέκτημα των frame-based επεξεργαστών είναι η αποφυγή συνηθισμένων λαθών, όπως το κλείσιμο συμβόλων τερματισμού σε μια δομή, αφού αυτό θεωρείται ενιαίο με το υπόλοιπο κομμάτι του κώδικα. Μεταφέρεται, δηλαδή ολόκληρο σε διαδικασίες αντιγραφής ή μετακίνησης σε συγκεκριμένες θέσεις, που θα επιλέξει ο χρήστης, χωρίς να υπάρχει ο κίνδυνος να ξεχαστεί κάποιο τμήμα κώδικα, το οποίο ανήκει μέσα σε αυτό το πλαίσιο.

Για τις προηγούμενες διαδικασίες, χρησιμοποιείται η τεχνική «σύρε και άφησε» («drag & drop»), ενώ το μετακινούμενο πλαίσιο δεν μπορεί να εναποτεθεί σε μη επιτρεπτές θέσεις μέσα στο ιεραρχικό δέντρο εντολών. Σταδιακά και άλλες εφαρμογές υιοθέτησαν αυτόν τον υβριδικό επεξεργαστή στα εκπαιδευτικά τους συστήματα, ώστε να διευκολύνουν τους νέους προγραμματιστές ή και να προσφέρουν μια γέφυρα μετάβασης από τους block-based επεξεργαστές στους κλασσικούς text-based επεξεργαστές.

2.3. Το προγραμματιστικό μοντέλο

Το προγραμματιστικό μοντέλο του Greenfoot αποτελείται από μια κλάση κόσμου «World class», που απεικονίζεται σε ένα παράθυρο της οθόνης με μια τετράγωνη περιοχή και πλαισιώνεται από έναν αριθμό αντικειμένων, τα οποία παίζουν τον ρόλο των ηθοποιών (actor objects), εμφανίζονται μέσα στον κόσμο και μπορούν να προγραμματιστούν ανεξάρτητα το ένα από το άλλο. Τόσο ο κόσμος, όσο και οι ηθοποιοί αντιπροσωπεύουν αντικείμενα της γλώσσας προγραμματισμού Java, τα οποία αντανακλώνται σε κλάσεις (class) αυτής της γλώσσας. Το Greenfoot προσφέρει μεθόδους (methods), δηλαδή συναρτήσεις που ανήκουν μέσα σε μια κλάση για τον εύκολο προγραμματισμό αυτών των ηθοποιών (actor) και περιλαμβάνει μεθόδους, που αφορούν στην κίνηση μέσα στον κόσμο, στην περιστροφή, στις αλλαγές στην εμφάνιση, στον έλεγχο συγκρούσεων με άλλους ηθοποιούς κτλ.

Ο προγραμματισμός μέσω του Greenfoot στην πιο απλή του μορφή αποτελείται από υπο-κλάσεις των δυο βασικών κλάσεων, που είναι η World (κόσμος) και Actor (ηθοποιός). Ένα στιγμιότυπο (instance) της υπο-κλάσης world αντιπροσωπεύει τον κόσμο, στον οποίο η «εκτέλεση» (run) του Greenfoot λαμβάνει χώρα. Οι υπο-κλάσεις των actor είναι αντικείμενα (objects), τα οποία υπάρχουν και δρουν μέσα στο world. Το στιγμιότυπο της κλάσης world δημιουργείται αυτόματα από το περιβάλλον εργασίας. Η «εκτέλεση» (run) του Greenfoot ενεργοποιεί έναν ενσωματωμένο (built-in) βρόγχο, που επανειλημμένως καλεί την μέθοδο act (το σημείο εισαγωγής για τη δράση του κάθε ηθοποιού) ξεχωριστά για κάθε έναν actor, που υπάρχει στον κόσμο. Αυτό σημαίνει ότι το προγραμματιστικό σενάριο αποτελείται κυρίως από την υλοποίηση (implementation) της μεθόδου act για κάθε έναν actor, που θα δράσει μέσα στον κόσμο. Η υλοποίηση πραγματοποιείται με τις κλασσικές εντολές της γλώσσας προγραμματισμού Java. Η διασύνδεση API (Application Programming Interface) των μεθόδων του Greenfoot με την εφαρμογή, που τις χρησιμοποιεί, προσφέρει μια σειρά από κοινές εργασίες, όπως η σχεδιοκίνηση (animation), ηχητική υποστήριξη, γεννήτριες τυχαίων αριθμών και διαχείριση εικόνων. Όλες οι βασικές (standard) βιβλιοθήκες τις Java μπορούν επίσης να χρησιμοποιηθούν σε συνδυασμό με το Greenfoot, ώστε να προκύψουν πιο περίπλοκα και πιο εκλεπτυσμένα αποτελέσματα.



```
public void act()  
{  
    setLocation (getX() + 4, getY());  
    // Moves the Actor 4 cells to the right  
    setRotation (getRotation() + 2);  
    // Rotates the Actor 2 degrees clockwise  
}
```

Εικόνα 5. Παράδειγμα υλοποίησης μιας απλής ACT μεθόδου (μετακινεί τον actor - μπάλα ποδοσφαίρου - τέσσερα βήματα αριστερά από την τρέχουσα θέση του και ταυτόχρονα τον περιστρέφει κατά δύο μοίρες επίσης αριστερά)

Αυτό το μοντέλο του Greenfoot έχει στόχο κυρίως να προσελκύσει τους αρχάριους προγραμματιστές, παρέχοντας εύκολη πρόσβαση στα κινούμενα γραφικά, στον ήχο και στην διάδραση μεταξύ των αντικειμένων. Το περιβάλλον εργασίας έχει υψηλό επίπεδο αλληλεπίδρασης και ενθαρρύνει τους εν δυνάμει εξερευνητές να πειραματιστούν. Παιδαγωγικά, ο σχεδιασμός του μοντέλου αυτού βασίζεται στον κονστρουκτιβισμό και τη μαθητεία (apprenticeship) υπό την παρακολούθηση κάποιου εκπαιδευτή. Επιπλέον, το περιβάλλον εργασίας σχεδιάστηκε, δίνοντας έμφαση στη γενική έννοια και φιλοσοφία του όρου «Αντικειμενοστραφής προγραμματισμός» (object-oriented programming). Έννοιες, όπως κλάση ή αντικείμενο, αλλά και οι σχέσεις που τα διέπουν, όπως μέθοδοι, παράμετροι και αλληλεπίδραση αντικειμένων εκφράζονται δια της οπτικοποιημένης αναπαράστασης και της καθοδηγούμενης αλληλεπίδρασης. Ο στόχος ήταν η δημιουργία και υποστήριξη ενός θεωρητικού μοντέλου, το οποίο δύναται να αντιπροσωπεύσει ρεαλιστικά μια σύγχρονη γλώσσα αντικειμενοστραφούς προγραμματισμού.

2.4. Υποστήριξη

Για τους χρήστες του Greenfoot υπάρχει μία κοινότητα, που ονομάζεται **Greenfoot Gallery** (<https://greenroom.greenfoot.org/>), η οποία παρέχει μία πλατφόρμα για τη δημοσίευση και συζήτηση των διαφόρων υλοποιήσεων, που πραγματοποιούνται με αυτό. Οποιοσδήποτε μπορεί να δημιουργήσει έναν λογαριασμό και να δημοσιεύσει την εργασία του. Όταν γίνει αυτό, τότε τα σενάρια μπορεί να «τρέξουν» (run) μέσα σε έναν απλό φυλλομετρητή (web browser) και αυτομάτως οποιοσδήποτε στον πραγματικό πλέον κόσμο να δει, να παίξει και αξιολογήσει την εργασία αυτή.

Η δυνατότητα της εύκολης δημοσίευσης ενός προγράμματος, φτιαγμένο στο Greenfoot, φαίνεται να αποτελεί σημαντικό κίνητρο για τους αρχάριους προγραμματιστές. Για τους εκπαιδευτές η πλατφόρμα παρέχει το Greenroom, δηλαδή μία κοινότητα στην οποία συζητιούνται εκπαιδευτικές στρατηγικές, ανταλλαγή εμπειριών και υλικού. Παράλληλα, το Greenroom παρέχει κοινή αποθήκη υλικού (shared repository) με εκπαιδευτικό υλικό, το οποίο περιλαμβάνει έναν μεγάλο αριθμό από φύλλα εργασίας, σχέδια ιδεών, διαφάνειες και άλλα εκπαιδευτικά βοηθήματα. Ο δικτυακός αυτός τύπος υποστηρίζεται, επίσης, από την εταιρία Oracle.

2.5. Εγκατάσταση του περιβάλλοντος Greenfoot

Η εγκατάσταση του Greenfoot είναι μια απλή και εύκολη διαδικασία. Ο χρόνος, που απαιτείται, εξαρτάται από την ταχύτητα της εκάστοτε σύνδεσης στο διαδίκτυο.



Εικόνα 6. Η αρχική σελίδα του ιστότοπου Greenfoot (www.greenfoot.org)

Στην επιλογή της σελίδας «The Software» δίνεται η δυνατότητα επιλογής λειτουργικού συστήματος, καθώς διατίθεται για Windows, Linux (Ubuntu) και Mac Os X. Στην περίπτωση των Windows απλά γίνεται εγκατάσταση του *.msi αρχείου, που «θα κατεβεί» από την επίσημη ιστοσελίδα του Greenfoot. Εάν υπάρχει προηγούμενη έκδοση της εφαρμογής, τότε θα προηγηθεί απεγκατάσταση με τη γνωστή διαδικασία από το Control Panel.

Στην περίπτωση του MacOS θα πρέπει πρώτα να «κατεβεί» το συμπιεσμένο αρχείο *.zip και μετά την αποσυμπίεσή του να μετακινηθεί στον φάκελο «Applications».

Τόσο σε περιβάλλον Windows, όσο και MacOS ο χρήστης είναι υπεύθυνος για την εγκατάσταση της τελευταίας έκδοσης της Java Platform.

Στην περίπτωση του Ubuntu Linux, θα πρέπει να «κατεβεί» το αρχείο από τον δικτυακό τόπο και να γίνει χρήση του Ubuntu Software Manager. Με αυτόν τον τρόπο, θα εγκατασταθεί αυτόματα και το OpenJDK 8 (Open Java Development Kit) μια ελεύθερη και ανοιχτού κώδικα υλοποίηση του Java Platform Standard Edition (Java SE). Παρόλο που το Greenfoot, κατά πάσα πιθανότητα, θα συνεργαστεί χωρίς προβλήματα και με αυτήν την έκδοση της Java, συστήνεται να γίνει χρήση της Oracle JDK 8 για αποφυγή πιθανών προβλημάτων.

Τέλος, για την περίπτωση του Debian Unix, θα πρέπει να γίνει πρώτα εγκατάσταση της Oracle JDK ή του OpenJDK δευτερευόντως (αν και δεν συστήνεται επίσης) και στη συνέχεια να εγκατασταθεί το πακέτο '*Greenfoot-linux-351.deb*' με την εντολή '*gdebi Greenfoot-linux-351.deb*'.

Για οποιοδήποτε άλλο λειτουργικό σύστημα μπορεί να γίνει η εγκατάσταση με το αρχείο Java, που φέρει την επέκταση *.jar. Θα πρέπει, ωστόσο, να προηγηθεί η εγκατάσταση της κατάλληλης πλατφόρμας Java (από την έκδοση 3 και μετά του Greenfoot, απαιτείται το JDK 8 μαζί με το JavaFx), αν και συστήνεται όπως και στις προηγούμενες περιπτώσεις η έκδοση της Oracle, που υπάρχει στη διεύθυνση <https://www.oracle.com/technetwork/java/javase/downloads/index.html>.

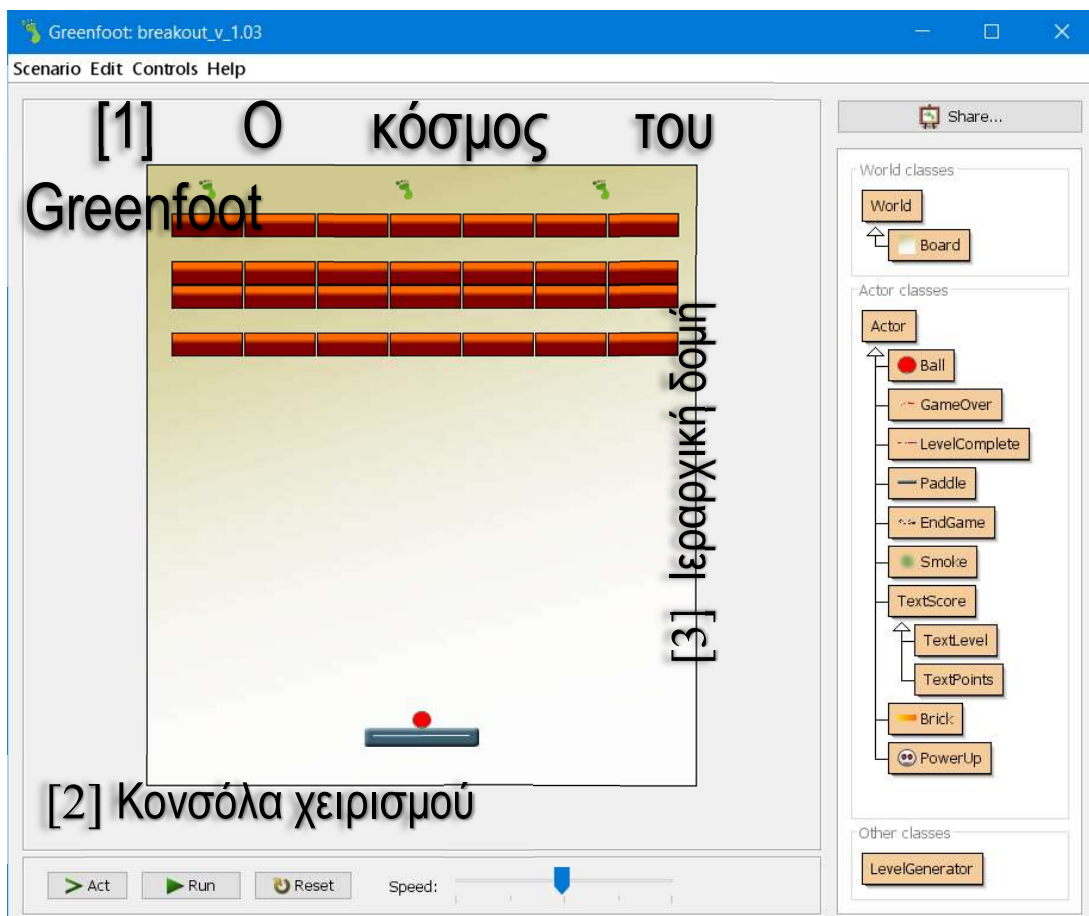
Στην τελευταία περίπτωση, απλά πληκτρολογείτε την εντολή παρακάτω σε ένα παράθυρο τερματικού (command line window) ή κάνετε διπλό κλικ στον διαχειριστή αρχείων επάνω στο όνομα του αρχείου, εφόσον υποστηρίζεται από το αντίστοιχο λειτουργικό σύστημα ('*java -jar Greenfoot-generic-351.jar*'). Ίσως χρειαστεί να δώσετε την πλήρη διεύθυνση του αρχείου java μέσα στο σύστημα καταλόγων του σκληρού σας δίσκου (full path), όπως για παράδειγμα '*/opt/jdk1.8.0_66/bin/java -jar Greenfoot-generic-351.jar*', αν το Oracle JDK εγκαταστάθηκε στον φάκελο '*/opt/jdk1.8.0_66*'.

Επίσης, υπάρχει μια ειδική έκδοση των Windows, όπου μπορεί να «τρέξει» σε USB stick και εμπεριέχεται και το JDK. Επισημαίνεται ότι για τους επεξεργαστές Intel και AMD από την έκδοση του 3.6.0 και μετά απαιτείται η 64bit έκδοση των αντίστοιχων επεξεργαστών και λειτουργικών συστημάτων. Αν απαιτείται να γίνει χρήση της 32bit έκδοσης, τότε θα πρέπει να γίνει εγκατάσταση της έκδοσης 3.5.4 του Greenfoot ή και παλαιότερη, από τη διεύθυνση που φαίνεται παρακάτω https://www.greenfoot.org/download_old.

2.6. Περιβάλλον εργασίας

Το Greenfoot αποτελείται από έναν "κόσμο" (world) και από "ηθοποιούς" (actors), οι οποίοι ενεργούν επάνω σε αυτόν. Ο κόσμος (αριθμός 1 στην εικόνα 7) καταλαμβάνει το μεγαλύτερο μέρος

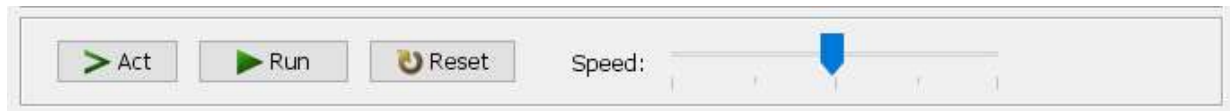
του παραθύρου και ουσιαστικά αποτελεί τη βάση, πάνω στην οποία ενεργούν οι ηθοποιοί. Εκεί «εκτελείται» (run) το πρόγραμμα και βλέπουμε τα αποτελέσματα των όποιων ενεργειών.



Εικόνα 7. Το παράθυρο και οι βασικές εργαλειοθήκες στο περιβάλλον Greenfoot

Δεξιά στο παράθυρο, εμφανίζεται η ιεραρχική δομή των κλάσεων (classes) (αριθμός 3 στην εικόνα 7), στην οποία οργανώνεται τόσο ο κόσμος, όσο και οι ηθοποιοί. Οι κλάσεις (classes) αποτελούν σύνολα αντικειμένων (objects) με συγκεκριμένες ιδιότητες (methods/attributes), τα οποία, όταν κληρονομούν ένα άλλο αρχικό σύνολο αντικειμένων (parent object inheritance), τότε υιοθετούν και εκείνες τις αρχικές ιδιότητες. Για παράδειγμα, σε μια κλάση «έπιπλο» μπορούμε να δημιουργήσουμε υπο-κλάσεις «τραπέζι» και «πολυθρόνα». Αυτές οι δύο υπο-κλάσεις θα έχουν όλες τις ιδιότητες της αρχικής οντότητας – κλάσης «έπιπλο» συν τις δικές τους. Κάνοντας κλικ σε μία κλάση, μπορούμε να την εισάγουμε χειροκίνητα (εναλλακτικά χρησιμοποιούμε κώδικα), να τη διαγράψουμε ή να προσθέσουμε κώδικα.

Στο κάτω μέρος του παραθύρου υπάρχει η κονσόλα χειρισμού (εικόνα 8), η οποία διαθέτει τις επιλογές:



Εικόνα 8. Κονσόλα χειρισμού

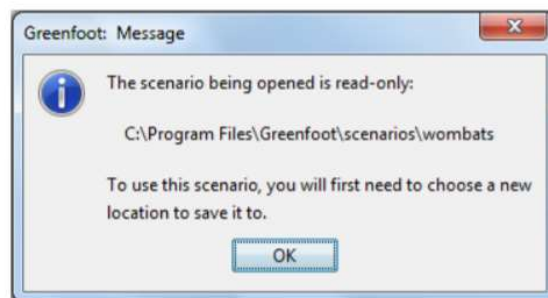
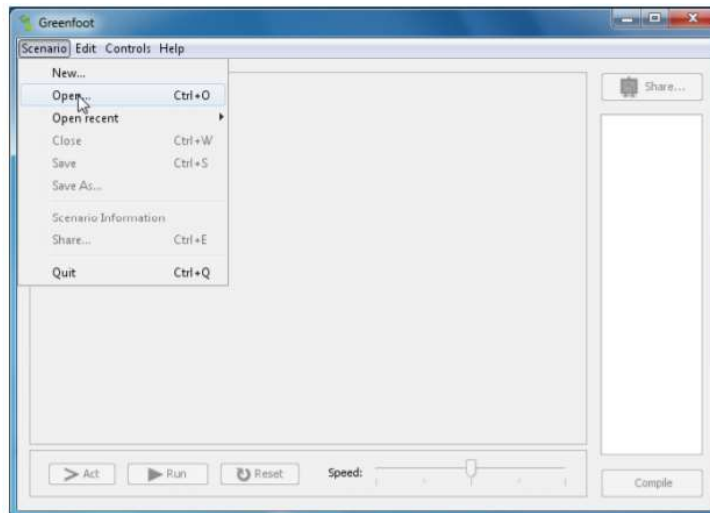
- **Act:** Βηματική «εκτέλεση» (run) των εντολών του σεναρίου.
- **Run:** «Εκτέλεση» (run) ολόκληρου του σεναρίου ανεξαιρέτως.
- **Reset:** Δημιουργία ενός νέου κόσμου από την αρχή, χωρίς τις παρεμβάσεις, που ενδεχομένως έγιναν από τον χρήστη.
- **Μπάρα κύλισης ταχύτητας (Speed):** Καθορισμός της ταχύτητας κίνησης των ηθοποιών (actors).

Σύμφωνα με τα παραπάνω, κάθε λειτουργικό σενάριο θα πρέπει να έχει τουλάχιστον έναν κόσμο και κάποιους ηθοποιούς (actors). Για να εκτελεστεί το σενάριο, θα πρέπει πρώτα να γίνει μεταγλώττιση (compile) αυτού. Εφόσον δεν προκύψουν σφάλματα, τότε το αποτέλεσμα της εκτέλεσης (run), θα εμφανιστεί μέσα στον κόσμο (αριθμός 1 στην εικόνα 7).

2.7. Παρουσίαση σεναρίου Greenfoot

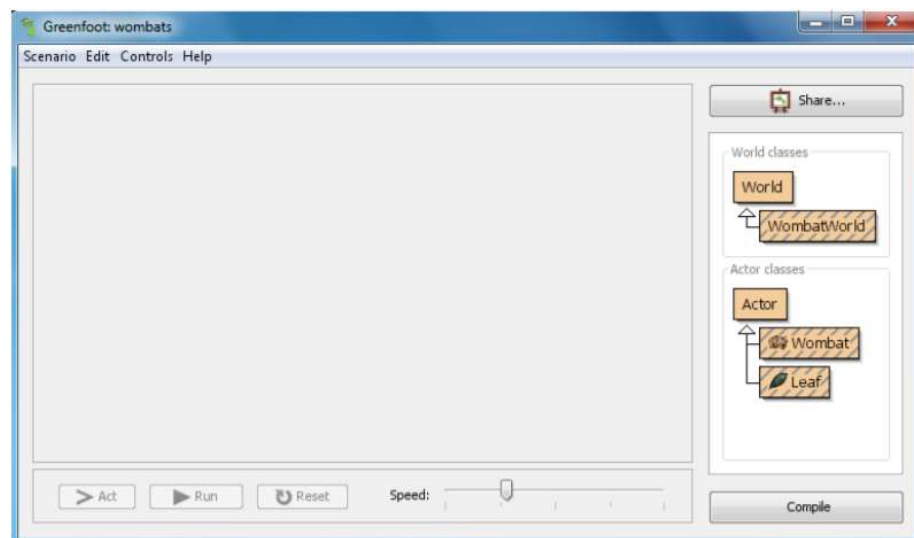
Προκειμένου να γίνει κατανοητό το περιβάλλον του Greenfoot και να παρουσιαστούν αναλυτικά ορισμένες από τις δυνατότητές του, θα ακολουθήσει η χρήση ενός έτοιμου σεναρίου, το οποίο διανέμεται δωρεάν με την εγκατάσταση του Greenfoot και φέρει το όνομα «Wombats» (Φασκολώμυς: μαρσιποφόρο φυτοφάγο ζώο, που ζει στην Αυστραλία). Η χρήση έτοιμου σεναρίου αποφασίστηκε, ώστε ο αναγνώστης να έχει άμεση πρόσβαση σε όλο το υλικό, που απαιτείται για την ολοκλήρωσή του.

Μέσα από το μενού «Scenario» και «Open», που υπάρχει μέσα στον φάκελο εγκατάστασης του Greenfoot και στον υποφάκελο «Scenarios», θα πρέπει να ανοιχτεί το αρχείο «Wombats». Αν δεν γίνει αλλαγή του φακέλου κατά την αρχική εγκατάσταση του Greenfoot, τότε αυτός βρίσκεται εξ ορισμού στη διαδρομή “C:\Program Files\Greenfoot\Scenarios\”, εφόσον το λειτουργικό σύστημα που χρησιμοποιείται, είναι τα Microsoft Windows. Το αρχείο-σενάριο θα ανοίξει μόνο για ανάγνωση (read only), οπότε για οποιαδήποτε αλλαγή απαιτείται πρώτα η αποθήκευσή του με ένα διαφορετικό όνομα.



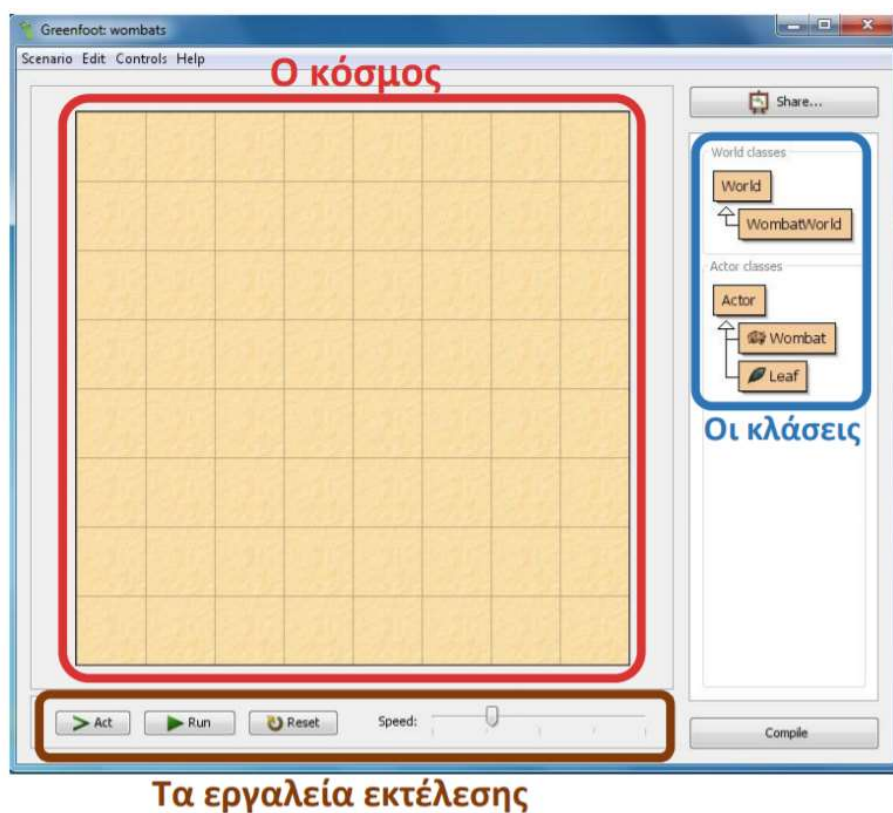
Εικόνα 9. Άνοιγμα αρχείου και αποθήκευση με άλλο όνομα

Αμέσως μετά την αποθήκευση του αρχείου θα εμφανιστούν οι κλάσεις «κόσμος» (world) και «ηθοποιοί» (actor) στην ιεραρχική δομή κλάσεων.



Εικόνα 10. Το αρχικό παράθυρο

Για να γίνει η εμφάνιση του κόσμου, ο οποίος αντιπροσωπεύεται από ένα πλέγμα (grid), θα πρέπει να προηγηθεί η μεταγλώττιση (compilation) του σεναρίου.



Εικόνα 11. Εμφάνιση του «κόσμου» μετά τη μεταγλώττιση



Εικόνα 12. Εισαγωγή αντικειμένου

Για την εισαγωγή αντικειμένων στον «νέο κόσμο», θα πρέπει να γίνει δεξί κλικ (για τους υπολογιστές Mac, ο αντίστοιχος συνδυασμός είναι Control-κλικ) στην κλάση «Wombat» από την ιεραρχική δομή των κλάσεων. Στο αναδυόμενο παράθυρο, που θα ανοίξει, θα πρέπει να επιλεγεί η «new Wombat()» και στη συνέχεια, απαιτείται ξανά κλικ οπουδήποτε μέσα στον «κόσμο».

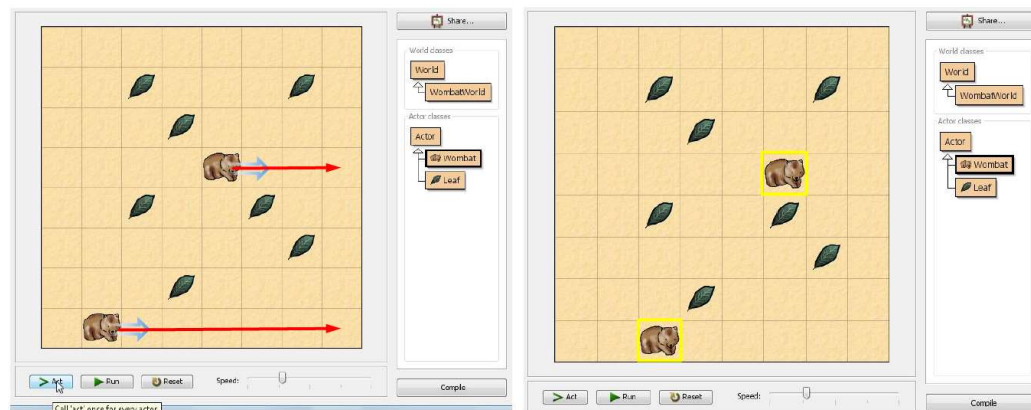
Αυτόματα θα εμφανιστεί ένα «wombat» στο σημείο, που έγινε το κλικ. Αυτό αποτελεί ένα στιγμιότυπο (instance) του αντικειμένου «wombat», το οποίο κληρονομεί όλες τις ιδιότητες του γονικού αντικειμένου «actor».

Για την εισαγωγή μερικών φύλλων μέσα στον κόσμο θα πρέπει να ακολουθηθεί η ίδια διαδικασία, μόνο που τώρα θα επιλεγεί η κλάση «Leaves». Κάνοντας επαναλαμβανόμενα κλικ, έχοντας ταυτόχρονα πατημένο το πλήκτρο **Shift** μέσα στον κόσμο, μπορούν να προστεθούν πολλαπλά στιγμιότυπα αυτού του αντικειμένου. Αντίθετα, για τη διαγραφή κάποιου αντικειμένου από τον κόσμο, θα πρέπει να γίνει δεξί κλικ επάνω στην εικόνα του και από το μενού, που θα ανοίξει, να επιλεγεί το «Remove».



Εικόνα 13. Διαγραφή αντικειμένου

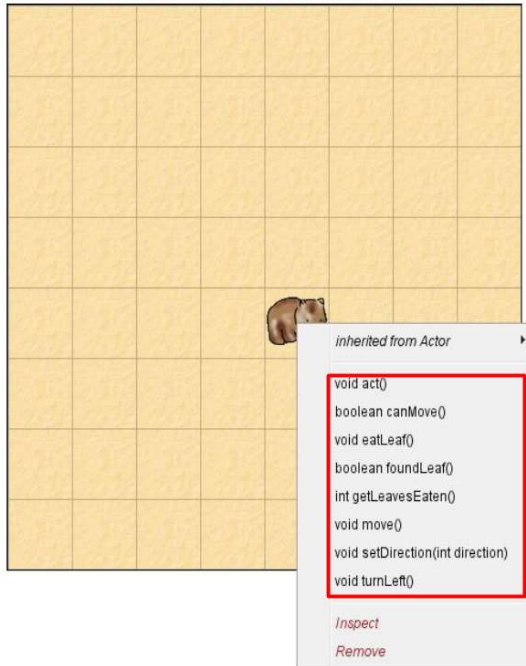
τα φύλλα είναι προγραμματισμένα να μην κάνουν τίποτα, ενώ αντίθετα τα «wombats» είναι προγραμματισμένα να κινούνται με τη φορά, που είναι τοποθετημένα. Αν τοποθετηθούν δύο ή περισσότερα «wombats» στον κόσμο και γίνει κλικ στο «Act» (βηματική εκτέλεση), θα κινηθούν όλα μαζί ταυτόχρονα προς την κατεύθυνση, που δείχνει το πρόσωπό τους. Στη συγκεκριμένη περίπτωση, κοιτούν προς τα δεξιά.



Εικόνα 14. Η λειτουργία «Act»

Εάν τα μικρά φυτοφάγα «wombats» συναντήσουν στον δρόμο τους κάποιο από τα διασκορπισμένα φύλλα μέσα στο πλέγμα του «κόσμου», τότε αυτό θα εξαφανιστεί, δίνοντας την αίσθηση ότι φαγώθηκε. Θα πρέπει να γίνει συνεχόμενο πάτημα του κουμπιού «Act», προκειμένου να δοθεί η ψευδαίσθηση της κίνησης. Διαφορετικά, το πάτημα του κουμπιού «Run» θα ενεργοποιήσει την εκτέλεση «run» του σεναρίου εσαεί, μέχρι ο χρήστης να το διακόψει, είτε με το «Reset», είτε με την παύση «Pause». Εφόσον το σενάριο εκτελείται, μπορεί να ρυθμιστεί η ταχύτητα εκτέλεσης με τον ρυθμιστή «Speed».

Επιπροσθέτως, ο χρήστης μπορεί να καλέσει κάποιες από τις μεθόδους, που συνδέονται με τον «actor». Μέθοδος ονομάζεται μία μοναδική πράξη, που ένα αντικείμενο μπορεί να εκτελέσει. Με δεξί



Εικόνα 15. Μέθοδοι, που αφορούν τον

κλικ επάνω σε ένα αντικείμενο και χωρίς να εκτελείται το σενάριο, μπορεί κανείς να δει τις διαθέσιμες μεθόδους (συναρτήσεις Java), που σχετίζονται με αυτό και να επιλέξει οποιαδήποτε από αυτές. Μπορεί να επιλεγεί μία μέθοδος, που θα ζητάει από το αντικείμενο να εκτελέσει μια πράξη, όπως για παράδειγμα να στρίψει αριστερά με την «turnLeft()» ή απλά να κινηθεί κατά μία θέση με την «move()».

Κάποιες άλλες μέθοδοι απλά επιστρέφουν μια απάντηση σε ερώτηση, που κάνει ο χρήστης, όπως για παράδειγμα η μέθοδος «getLeavesEaten()», που επιστρέφει τον αριθμό των φύλλων, που το «wombat» έχει καταναλώσει μέχρι στιγμής.

Όπως θα ανέμενε κανείς, μπορεί να εντοπίσει και τη μέθοδο «act()», η οποία καλείται κάθε φορά, που ο χρήστης κάνει κλικ στο αντίστοιχο κουμπί από τα εργαλεία εκτέλεσης ή διαδοχικά και εσαεί από το κουμπί της εκτέλεσης. Εδώ πλέον υπάρχει η επιλογή της κλήσης μόνο για τον επιλεγμένο ηθοποιό (actor).

Αν έχουν δημιουργηθεί πολλά αντικείμενα στον κόσμο, τα οποία δεν χρειάζονται πλέον ή υπάρχει η επιθυμία να ξεκινήσει κανείς από την αρχή, ένας εύκολος τρόπος, για να καταργηθεί ο υπάρχον «κόσμος» και να δημιουργηθεί ένας νέος, είναι με την επιλογή του κουμπιού «Reset» από τα εργαλεία εκτέλεσης. Ο «παλιός κόσμος» συμπεριλαμβανομένων και όλων των αντικειμένων, που υπήρχαν σε αυτόν, παύει να υπάρχει και δημιουργείται ένας νέος κενός «κόσμος» χωρίς όμως να διαγραφούν και οι κλάσεις. Αξίζει να σημειωθεί πως μπορεί να υπάρχει μόνο ένας «κόσμος» ανά σενάριο.

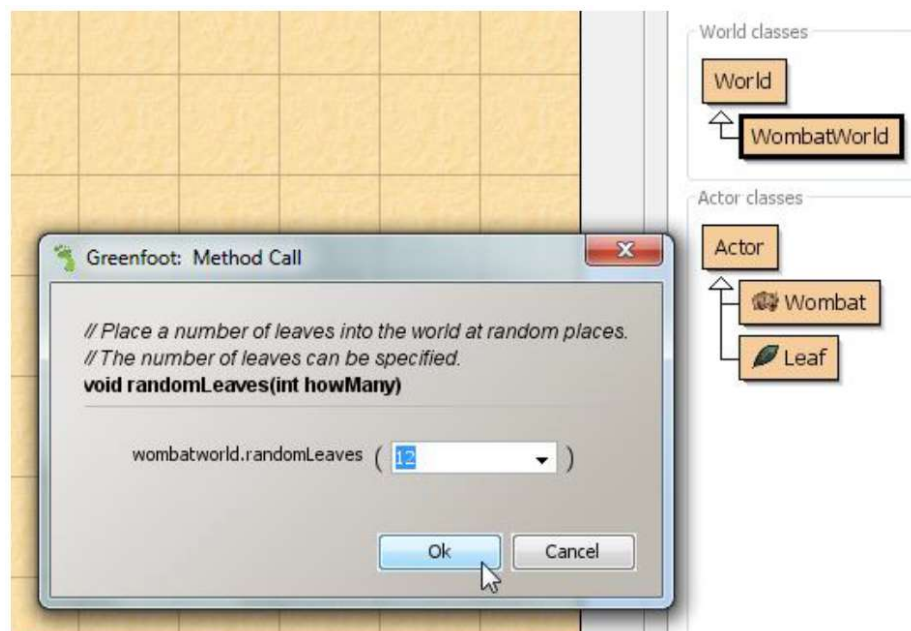
Όπως και οι ηθοποιοί, ο «κόσμος» έχει τις δικές του μεθόδους, που μπορούν να εμφανιστούν με δεξί κλικ επάνω σε αυτόν, χωρίς, ωστόσο, ο δείκτης του ποντικιού να ακουμπάει κάποιον ηθοποιό (actor).

Η μέθοδος «populate()» δημιουργεί τυχαίο αριθμό «wombats» και φύλλων (leaves) και τα τοποθετεί μέσα στον «κόσμο». Με αυτόν τον τρόπο, ο χρήστης μπορεί άμεσα να «εκτελέσει» το σενάριο χωρίς να μεσολαβήσει η χειροκίνητη τοποθέτηση ηθοποιών μέσα στον κόσμο, όπως αναφέρθηκε προηγουμένως.



Εικόνα 16. Η μέθοδος «populate()»

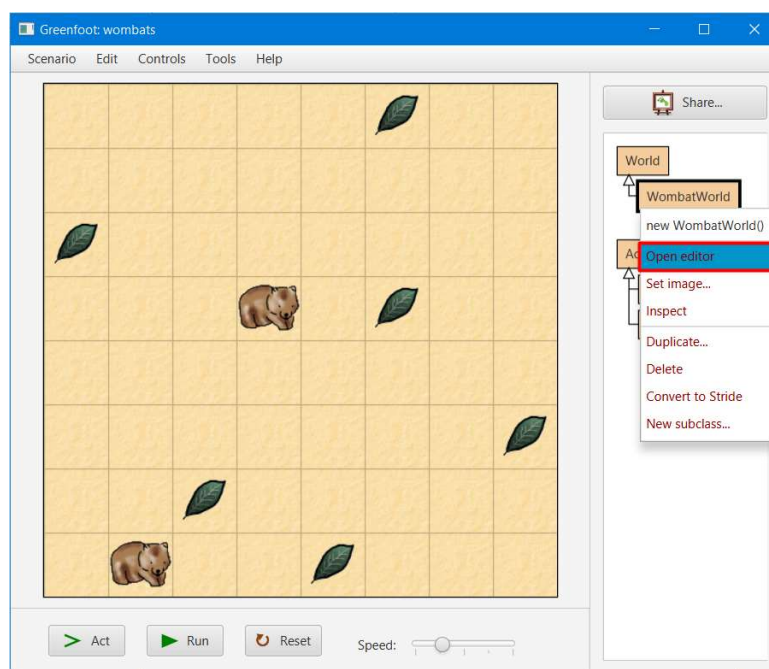
Μια άλλη μέθοδος του κόσμου, η «randomLeaves(int howMany)» τοποθετεί τυχαία έναν συγκεκριμένο αριθμό φύλλων μέσα στον «κόσμο». Ο αριθμός φύλλων δίνεται με την παράμετρο «howMany». Αν γίνει κλήση αυτής της μεθόδου, τότε ένα παράθυρο διαλόγου θα προτρέψει τον χρήστη να πληκτρολογήσει μια ακέραια τιμή, η οποία θα περάσει ως παράμετρος στη μέθοδο αυτή. Επειδή η τοποθέτηση των φύλλων είναι τυχαία, μπορεί δυο ή περισσότερα φύλλα να τοποθετηθούν στην ίδια θέση.



Εικόνα 17. Η μέθοδος «randomLeaves(int howMany)»

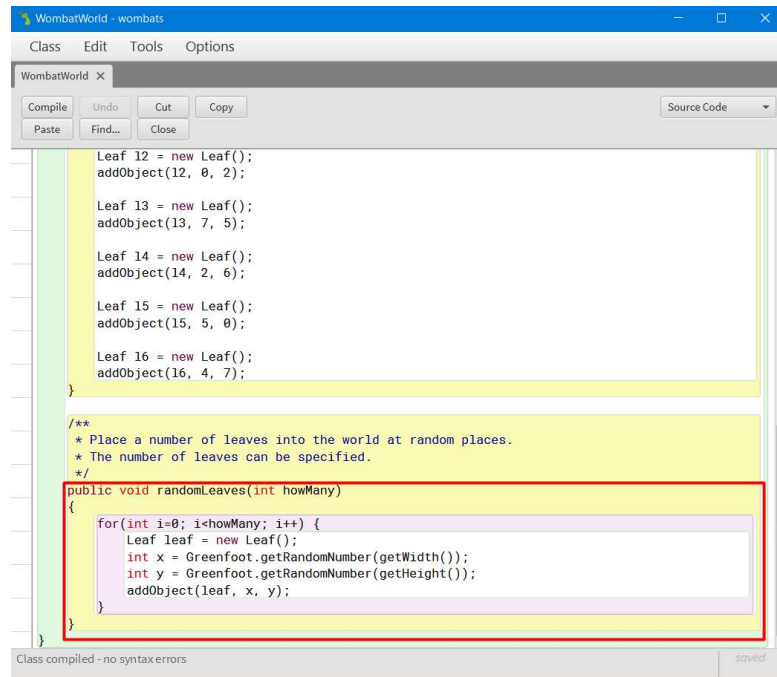
Όταν ο χρήστης εξοικειωθεί με τις βασικές λειτουργίες του περιβάλλοντος εργασίας, μπορεί να πειραματιστεί και με το προγραμματιστικό τμήμα της πλατφόρμας δια της εισαγωγής εντολών Java στον ενσωματωμένο επεξεργαστή κειμένου.

Για παράδειγμα, προκειμένου να έχει κανείς πρόσβαση στον κώδικα της μεθόδου «randomLeaves(int howMany)», θα πρέπει να κάνει δεξί κλικ επάνω στο «WombatWorld» στο πάνελ των κλάσεων, που αποτελεί και υπο-κλάση της αρχικής «World», και στη συνέχεια να επιλέξει «Open editor».



Εικόνα 18. Open Editor

Στο κάτω μέρος του επεξεργαστή κειμένου φαίνεται η συγκεκριμένη συνάρτηση, η οποία σε έναν βρόγχο for καλεί τις συναρτήσεις «Greenfoot.getRandomNumber()» και «addObject()», όσες φορές δήλωσε ο χρήστης μέσω της παραμέτρου παραπάνω.



Εικόνα 19. Ο κώδικας της μεθόδου «randomLeaves(int howMany)»

Με τον ίδιο ακριβώς τρόπο μπορεί κανείς να δει τον αντίστοιχο κώδικα σε Java για όλους τους ηθοποιούς. Ο χρήστης-προγραμματιστής μπορεί να παρέμβει στον κώδικα, προσθέτοντας ή διαγράφοντας στοιχεία και εντολές, με σκοπό να αλλάξει τον τρόπο εκτέλεσης του παιχνιδιού. Μπορεί να γίνει χρήση μεθόδων (συναρτήσεων), που παρέχει το ίδιο το Greenfoot, αλλά και κλασικών μεθόδων, τις οποίες χρησιμοποιεί η ίδια η Java. Τα αποτελέσματα οπτικοποιούνται άμεσα στον κόσμο του Greenfoot, με αποτέλεσμα να διατηρείται αμείωτο το ενδιαφέρον του χρήστη.

Για παράδειγμα, κατά την αρχικοποίηση του κόσμου λίγο παραπάνω από τον πηγαίο κώδικα, που χρησιμοποιήθηκε προηγουμένως, υπάρχει η μέθοδος «WombatWorld()». Η εντολή «super()» δημιουργεί έναν νέο «κόσμο» πλέγματος (grid) με κελιά, που έχουν διάσταση 8x8 και κάθε ένα από αυτά 60x60 εικονοστοιχεία (pixels). Ακριβώς από κάτω, υπάρχει η μέθοδος «setBackground()», η οποία δέχεται ως παράμετρο ένα αρχείο εικόνας.

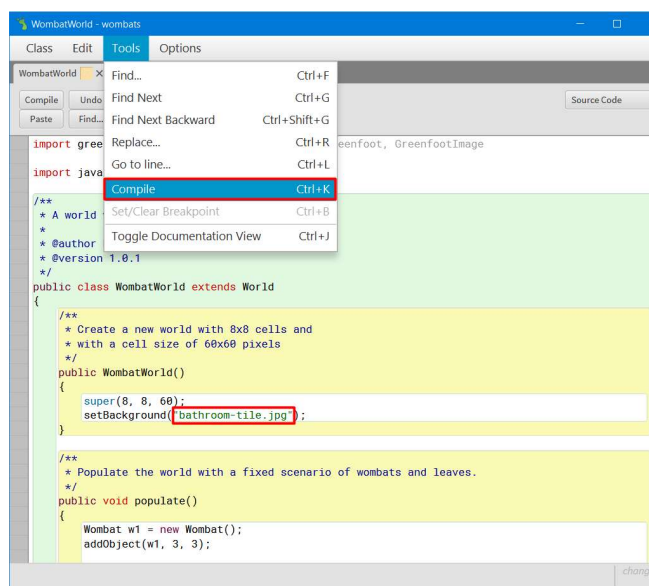
```

public class WombatWorld extends World
{
    /**
     * Create a new world with 8x8 cells and
     * with a cell size of 60x60 pixels
     */
    public WombatWorld()
    {
        super(8, 8, 60);
        setBackground("cell.jpg");
    }
}

```

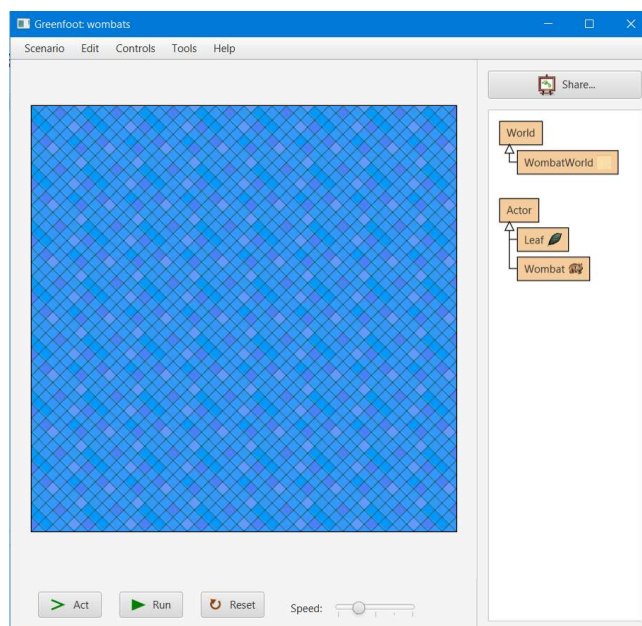
Εικόνα 20. Η εντολή «super()» και η μέθοδος «setBackground()»

Μέσα στον φάκελο «wombats» και στον υποφάκελο «images» υπάρχει η εικόνα «bathroom-tile.jpg». Αντικαθιστώντας την αρχική «cell.jpg» με την «bathroom-tile.jpg», αυτόματα θα αλλάξει η εικόνα του «κόσμου». Για την αποφυγή συντακτικών ή ορθογραφικών σφαλμάτων συστήνεται να γίνει πρώτα ένας έλεγχος μέσω του μενού «Tools» και της επιλογής «Compile» ή απευθείας με τον συνδυασμό πλήκτρων Ctrl+K.



Εικόνα 21. Αλλαγή αρχείου με χρήση κώδικα

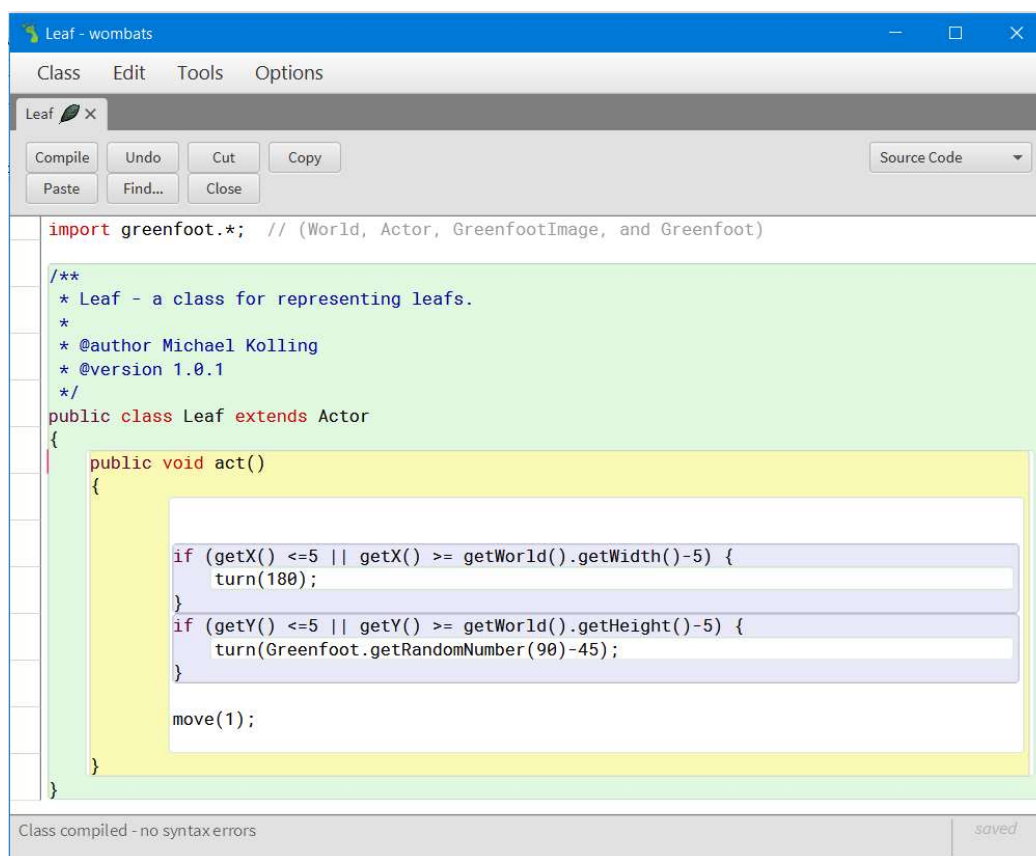
Εφόσον δεν υπάρξουν σφάλματα και κλείσει το παράθυρο του κώδικα, αυτόματα θα εμφανιστεί η νέα εικόνα στον «κόσμο».



Εικόνα 22. Ο νέος "Κόσμος"

Ο αντίστοιχος κώδικας, που αφορά τα φύλλα «Leaf», είναι σχεδόν άδειος. Εμφανίζεται μόνο η νέα υπο-κλάση «Leaf», η οποία κληρονομεί τις ιδιότητες της γονικής κλάσης «Actor» με τη δήλωση «extends Actor». Αν μέσα σε αυτήν την κλάση μπει μια μέθοδος «Act», τότε τα φύλλα θα αποκτήσουν ενέργεια.

Για παράδειγμα, ο κώδικας παρακάτω (εικόνα 23) μετακινεί με την εντολή «move» κατά μία θέση δεξιά τα φύλλα. Όταν φτάσουν κοντά στα όρια του «κόσμου», γίνονται τυχαίες περιστροφές. Η θέση ενός φύλλου δίνεται με τις μεθόδους «getX()» και «getY()», ενώ τα όρια του «κόσμου», δίνονται με τις μεθόδους «getWorld().getWidth()» και «getWorld().getHeight()». Η περιστροφή γίνεται με τη μέθοδο «turn()», που λαμβάνει ως παράμετρο έναν αριθμό σε μοίρες. Τέλος, υπάρχει μια γεννήτρια τυχαίων αριθμών, που δεν είναι άλλη από τη μέθοδο «Greenfoot.getRandomNumber()», η οποία δέχεται ως παράμετρο έναν ακέραιο αριθμό και παράγει έναν τυχαίο αριθμό από το μηδέν έως τον αριθμό της παραμέτρου.



```
import greenfoot.*; // (World, Actor, GreenfootImage, and Greenfoot)

/**
 * Leaf - a class for representing leafs.
 *
 * @author Michael Kolling
 * @version 1.0.1
 */
public class Leaf extends Actor
{
    public void act()
    {
        if (getX() <=5 || getX() >= getWorld().getWidth()-5) {
            turn(180);
        }
        if (getY() <=5 || getY() >= getWorld().getHeight()-5) {
            turn(Greenfoot.getRandomNumber(90)-45);
        }

        move(1);
    }
}
```

Εικόνα 23. Δίνοντας κίνηση στα «φύλλα»

Όπως έχει ήδη αναφερθεί, σε κάθε αλλαγή του κώδικα απαιτείται και η μεταγλώττισή του (compilation). Το αποτέλεσμα της εκτέλεσης αυτής της προσθήκης είναι τα φύλλα να κινούνται

σχεδόν με τυχαίο τρόπο, δυσκολεύοντας τα μικρά «wombats» να τα φάνε και προσδίδοντας ένα μικρό επίπεδο δυσκολίας.

Τα «wombats» κινούνται ανεξάρτητα από έναν αυτοματοποιημένο αλγόριθμο μέσα στη δική τους κλάση, η οποία κληρονομεί τις ιδιότητες της γονικής κλάσης «Actor». Εστιάζοντας στη μέθοδο «act», μπορεί να παρατηρήσει κανείς τις βασικές κλήσεις των μεθόδων «eatLeaf()», «move()», και «turnLeft()».

```
/**
 * Do whatever the wombat likes to do just now.
 */
public void act()
{
    if(foundLeaf()) {
        eatLeaf();
    }
    else if(canMove()) {
        move();
    }
    else {
        turnLeft();
    }
}
```

Εικόνα 24. Οι μέθοδοι «eatLeaf()», «move()», και «turnLeft()»

Η μέθοδος «move()» μετακινεί το «wombat» κατά μία θέση ανάλογα με την κατεύθυνση, που έχει κατά τη στιγμή εκτέλεσης του κώδικα, και η οποία υποδηλώνεται με τη μεταβλητή «direction». Οι γενικότερες κατευθύνσεις δηλώνονται ως σταθερές ακριβώς παραπάνω.

```
private static final int EAST = 0;
private static final int WEST = 1;
private static final int NORTH = 2;
private static final int SOUTH = 3;

private int direction;
private int leavesEaten;
```

Εικόνα 25. Οι γενικές κατευθύνσεις

Αν το «wombat» δεν μπορεί να κινηθεί περαιτέρω, επειδή έχει φτάσει στα όρια του «κόσμου» (μέθοδος «canMove()»), τότε γίνεται μία στροφή πολλαπλάσια των 90 μοιρών ανάλογα με την τρέχουσα κατεύθυνση, που έχει εκείνη τη στιγμή (μεταβλητή «direction»).

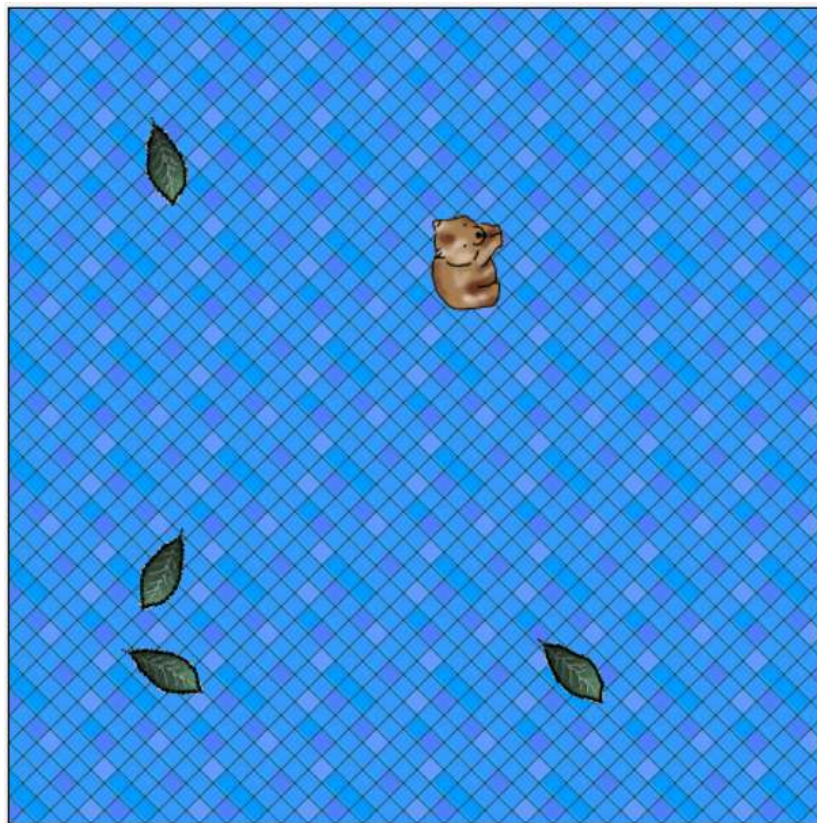
Αν αντικατασταθεί ο κώδικας μέσα στην «Act», ώστε οι κατευθύνσεις να δίνονται ανάλογα με τα πλήκτρα κατεύθυνσης του πληκτρολογίου, τότε ο χρήστης θα έχει τον πλήρη έλεγχο του ηθοποιού. Επιπλέον, γίνεται προαιρετικά και η χρήση της μεθόδου «setDirection()», ώστε η εικόνα του μικρού «wombat» να περιστραφεί αναλόγως.

```

* Do whatever the wombat likes to to just now.
*/
public void act()
{
    if(foundLeaf()) {
        eatLeaf();
    }
    else if(Greenfoot.isKeyDown("left")) {
        direction = WEST;
        move();
    }
    else if(Greenfoot.isKeyDown("right")) {
        direction = EAST;
        move();
    }
    else if(Greenfoot.isKeyDown("up")) {
        direction = NORTH;
        move();
    }
    else if(Greenfoot.isKeyDown("down")) {
        direction = SOUTH;
        move();
    }
    setDirection(direction);
}

```

Εικόνα 26. Χρήση πλήκτρων κατεύθυνσης

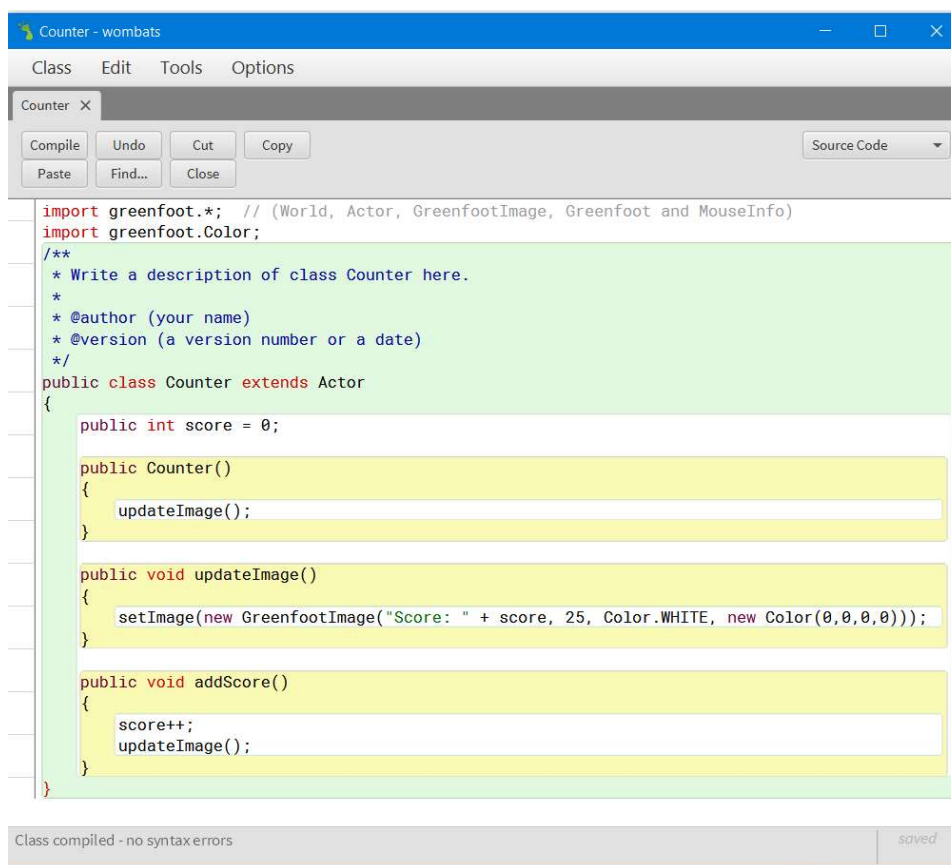


Εικόνα 27. Αλλαγή κατεύθυνσης (direction)

Τέλος, το παιχνίδι μπορεί να ολοκληρωθεί, εμφανίζοντας το σκορ του παίκτη σε κάποιο σημείο μέσα στον «κόσμο». Αν και υπάρχουν αρκετοί τρόποι εμφάνισης κειμένου ή εικόνας με δυναμικό τρόπο μέσα σε αυτόν, υπάρχει ένα δύσκολο σημείο, που θα πρέπει να ξεπεραστεί. Αυτό είναι ότι τόσο

το κείμενο, όσο και η εικόνα, εφόσον εμφανιστούν μέσα στον «κόσμο», δεν είναι δυνατόν να διαγραφούν, ώστε να ενημερωθούν με κάτι διαφορετικό, όπως η αλλαγή του σκορ. Αυτό έχει ως αποτέλεσμα να πέφτει η μία επάνω στην άλλη, δημιουργώντας σύγχυση και μπερδεμα.

Η λύση σε αυτό το πρόβλημα είναι η δημιουργία ενός ακόμη ηθοποιού, που θα παίζει το ρόλο του σκορ. Κάνοντας δεξί κλικ επάνω στην κλάση γονέα «Actor» και την επιλογή «New subclass...», δημιουργείται μία νέα κλάση, στην οποία μπορεί κανείς να δώσει όποιο όνομα επιθυμεί. Στο παράδειγμα αυτό, δόθηκε το όνομα «Counter». Μέσα στην κλάση υλοποιήθηκαν οι μέθοδοι «Counter()», «UpdateImage()» και «addScore()».



```
import greenfoot.*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)
import greenfoot.Color;

/**
 * Write a description of class Counter here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class Counter extends Actor
{
    public int score = 0;

    public Counter()
    {
        updateImage();
    }

    public void updateImage()
    {
        setImage(new GreenfootImage("Score: " + score, 25, Color.WHITE, new Color(0,0,0,0)));
    }

    public void addScore()
    {
        score++;
        updateImage();
    }
}
```

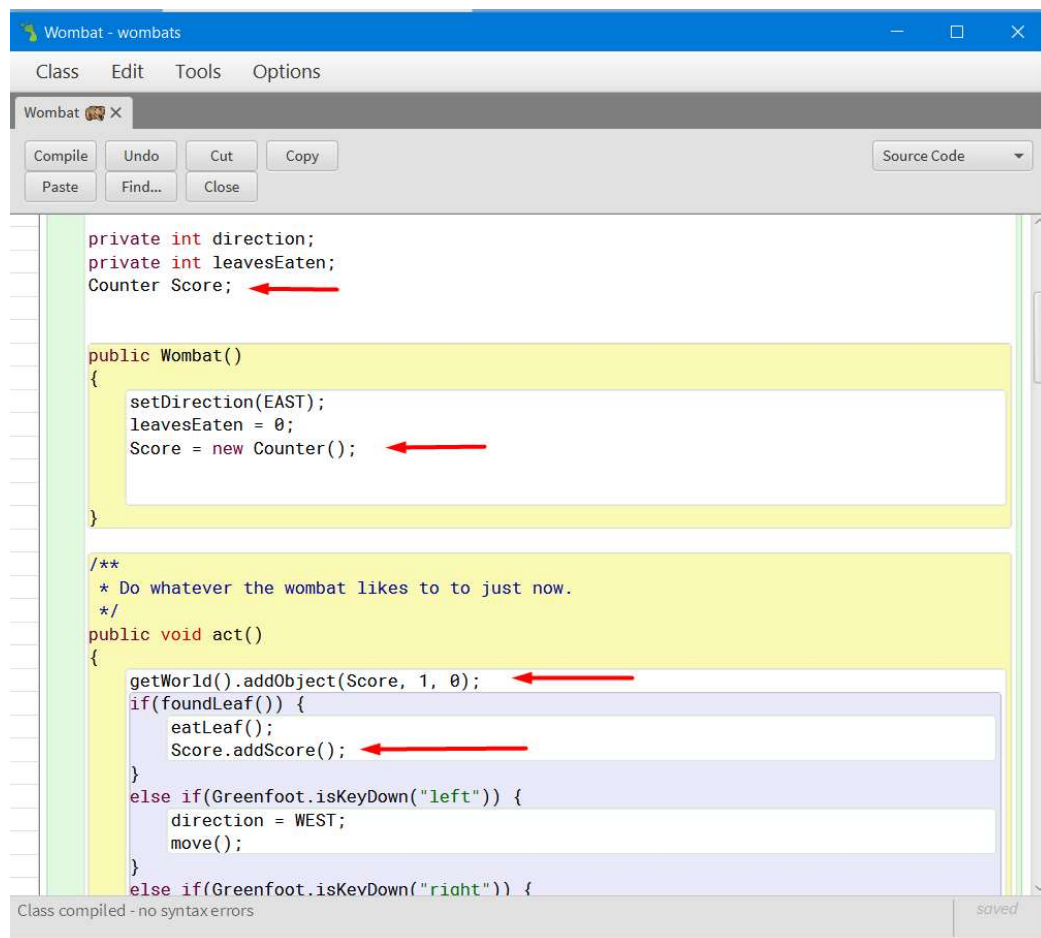
Εικόνα 28.Οι μέθοδοι «Counter()», «UpdateImage()» και «addScore()»

Η πρώτη μέθοδος, δηλαδή η «Counter()», αποτελεί την αρχικοποίηση (constructor) της νέας κλάσης. Ο ήρωας-ηθοποιός, που ελέγχει ο παίκτης, καλεί μέσα από μια δική του μέθοδο την «addScore()», η οποία απλά αυξάνει έναν μετρητή και τον εμφανίζει στην οθόνη με την «UpdateImage()».

Μέσα στις μεθόδους του «wombat» προστέθηκαν τέσσερις νέες εντολές. Η πρώτη κάνει τη γενική δήλωση της κλάσης «Counter». Η δεύτερη, μέσα στην αρχικοποίηση (constructor) της κλάσης

«wombat», δημιουργεί δυναμικά μέσα από το πρόγραμμα και την εντολή «new», αντί αυτό να γίνει με το ποντίκι από το τμήμα των κλάσεων, όπως αναφέρθηκε στην αρχή του κεφαλαίου.

Στη συνέχεια με την «addObject()» εμφανίζεται ο «actor» μέσα στον «κόσμο» στη θέση (1,0) μέσα στο πλέγμα δηλαδή 2^η στήλη και 1^η γραμμή (Οι γραμμές και οι στήλες ξεκινούν με την τιμή μηδέν).



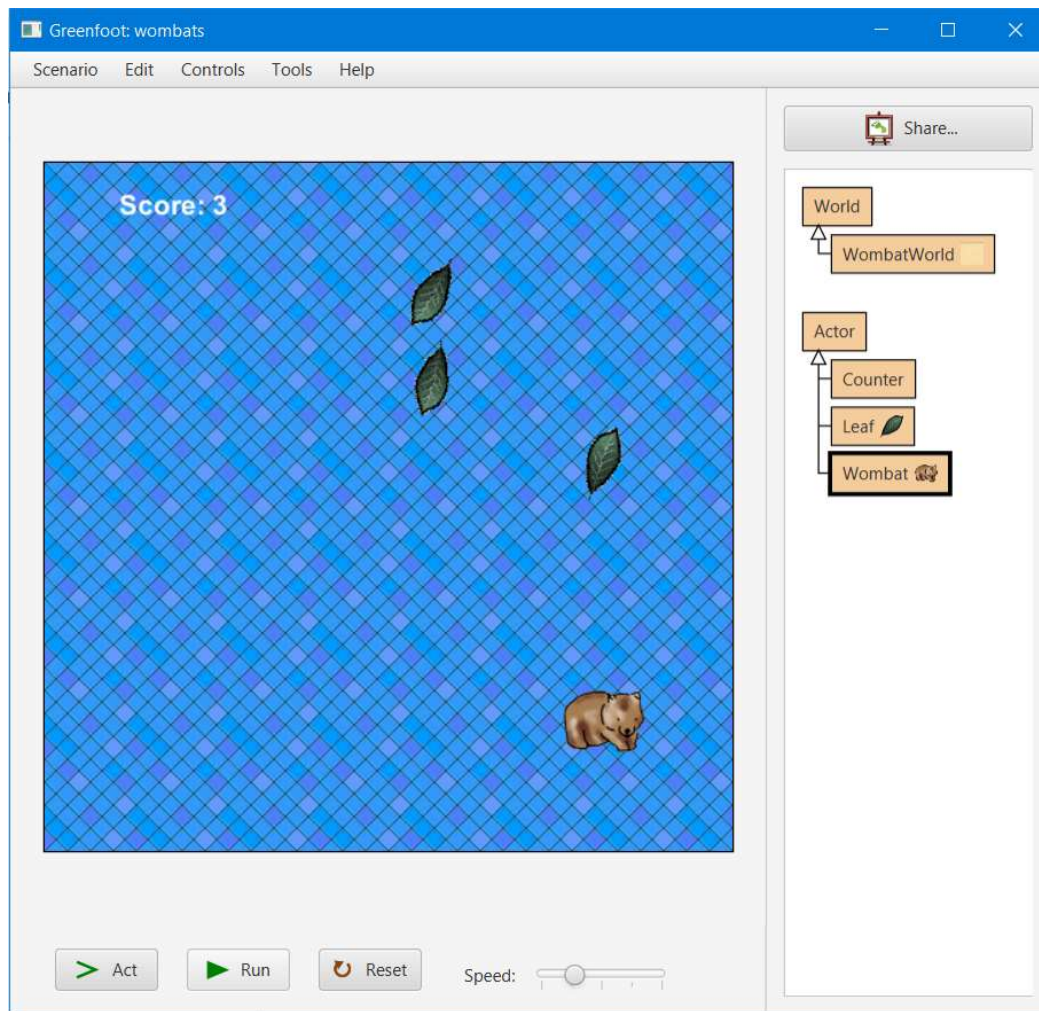
```
private int direction;
private int leavesEaten;
Counter Score;

public Wombat()
{
    setDirection(EAST);
    leavesEaten = 0;
    Score = new Counter();

    /**
     * Do whatever the wombat likes to do just now.
     */
    public void act()
    {
        getWorld().addObject(Score, 1, 0);
        if(foundLeaf()) {
            eatLeaf();
            Score.addScore();
        }
        else if(Greenfoot.isKeyDown("left")) {
            direction = WEST;
            move();
        }
        else if(Greenfoot.isKeyDown("right")) {
```

Εικόνα 29. Μέσα στις μεθόδους του «wombat» προστέθηκαν τέσσερις νέες εντολές

Τέλος, κάθε φορά που το μικρό «wombat» τρώει ένα φύλλο, δηλαδή μέσα στη δομή ελέγχου «if (foundLeaf())», τότε καλείται η μέθοδος «addScore()» της κλάσης «Counter», η οποία με τη σειρά της αυξάνει μια τοπική μεταβλητή, που αντιπροσωπεύει το σκορ και εν συνεχεία, εμφανίζει το περιεχόμενο αυτής της μεταβλητής μαζί με το μήνυμα «Score:» στην οθόνη.



Εικόνα 30. Εμφάνιση του «score»

ΜΕΡΟΣ Β: ΕΡΕΥΝΗΤΙΚΟ ΠΛΑΙΣΙΟ

ΕΙΣΑΓΩΓΗ

Στο πλαίσιο της παρούσας εργασίας υλοποιήθηκε ένα ρεαλιστικό εκπαιδευτικό σενάριο στην τάξη (Παράρτημα Ι) με συμμετέχοντες 107 ενήλικες εκπαιδευόμενους του 1^{ου} Δημόσιου Ινστιτούτου Επαγγελματικής Κατάρτισης (ΔΙΕΚ) Λάρισας και στη συνέχεια, έγινε συλλογή δεδομένων με χρήση ερωτηματολογίου (Παράρτημα ΙΙ).

Το θεωρητικό υπόβαθρο των εκπαιδευόμενων, ως προς τη χρήση Η/Υ και τον προγραμματισμό γενικότερα, ήταν κατά πλειοψηφία σε αρκετά ικανοποιητικό επίπεδο, αλλά με διακυμάνσεις ως προς την αποτελεσματικότητα, τη μεθοδολογία ανάλυσης και επίλυσης προβλημάτων και τον τρόπο υλοποίησης αυτών. Η εμπειρία τους, ειδικά στον προγραμματισμό, ήταν περιορισμένη, διότι μεγάλο μέρος αυτών δεν είχε ασχοληθεί ουσιαστικά στο παρελθόν και ελάχιστοι από το σύνολο των εκπαιδευόμενων ήταν ικανοποιημένοι με την παρακολούθηση των εργαστηριακών μαθήματων αρχιτεκτονικής υπολογιστών, βασισμένα σε γλώσσα προγραμματισμού Assembly του επεξεργαστή MOS Technology 6520, τα οποία ήταν περιορισμένου εύρους, αλλά μεγάλης εξειδίκευσης. Ωστόσο, οι εκπαιδευόμενοι κατείχαν βασικές γνώσεις αλγοριθμικής, είτε από τα Λυκειακά μαθήματα προηγούμενων ετών, είτε από τον κύκλο σπουδών του ΔΙΕΚ. Η γλώσσα προγραμματισμού Java, η οποία συγκαταλέγεται στις γλώσσες υψηλού επιπέδου και στην οποία στηρίζεται η εφαρμογή «Greenfoot», ήταν οικεία στην πλειοψηφία των συμμετεχόντων. (Οι πληροφορίες αυτές δόθηκαν από τον καθηγητή Πληροφορικής κ. Παπαϊωάννου Γεώργιο, ο οποίος γνώριζε το υπόβαθρο των εκπαιδευόμενων και εφάρμοσε το σενάριο μαθήματος, που του παρασχέθηκε).

Κύριος στόχος, ο οποίος μάλιστα τέθηκε εξ αρχής, ήταν η κατανόηση ενός θεωρητικού προγραμματιστικού αντικειμένου, που αφορούσε τον αντικειμενοστραφή προγραμματισμό (object oriented programming). Μέσω της προηγούμενης εμπειρίας τους, οι συμμετέχοντες γνώριζαν τη γενική φιλοσοφία βασικών προγραμματιστικών δομών, αλλά με έναν εντελώς πρωτογενή διαδικαστικό τρόπο προγραμματισμού, που αποτελούνταν από διακλαδώσεις μέσω των εντολών call, δηλαδή κλήση διαδικασιών και συναρτήσεων, ή jump από ετικέτα σε ετικέτα, σε συνδυασμό με τις συγκρίσεις καταχωρητών και τον έλεγχο του process register του ρεπερτορίου εντολών του επεξεργαστή 6502, κάτι, όμως, που δεν είχε πολύ μεγάλη σχέση με τις αντίστοιχες εντολές υψηλού επιπέδου - πολύ δε περισσότερο με τον αντικειμενοστραφή προγραμματισμό. (Οι πληροφορίες αυτές δόθηκαν από τον καθηγητή Πληροφορικής κ. Παπαϊωάννου Γεώργιο, ο οποίος γνώριζε το υπόβαθρο των εκπαιδευόμενων και εφάρμοσε το σενάριο μαθήματος, που του παρασχέθηκε).

Επόμενος στόχος ήταν η διατήρηση του ενδιαφέροντος των εκπαιδευόμενων καθ' όλη τη διάρκεια του μαθήματος, αλλά και μελλοντικά, η παρότρυνση δημιουργίας νέων εκπαιδευτικών σεναρίων, τα οποία θα βελτίωναν το δημιουργηθέν πρόγραμμα.

Τρίτος στόχος ήταν η εξοικείωση και κατανόηση βασικών όρων και εννοιών του προγράμματος, σχετικών με τα αντικείμενα, όπως κλάσεις, στιγμιότυπα κλάσεων, κληρονομικότητα, ιεραρχία.

Προφανώς, δεν έγινε πλήρης ανάλυση των δυνατοτήτων του αντικειμενοστραφούς προγραμματισμού, όπως αφηρημένες κλάσεις, υπερφόρτωση μεθόδων, ενθυλάκωση κτλ, λόγω του περιορισμένου διαθέσιμου χρόνου. Άλλωστε, ουσιαστική γνώση ενός αντικειμένου σημαίνει σύνδεση και αξιοποίηση με την προγενέστερη γνώση, ώστε να παραχθεί νέα σε συνδυασμό με την επίλυση σχετικών προβλημάτων.

Για την υλοποίηση του πειράματος τόσο η διδασκαλία, όσο και η έρευνα διενεργήθηκαν σε δύο εξάμηνα σπουδών (Α' εξάμηνο - Οκτώβριος 2018 και Β' εξάμηνο - Μάρτιος 2019), προκειμένου να συμπληρωθεί ένας ικανός αριθμός συμμετεχόντων. Η διδασκαλία ολοκληρώθηκε σε έναν κύκλο 8 μαθημάτων με διάρκεια μιάμισης ώρας το καθένα. Συνολικά, χρειάστηκε το πέρας ενός μήνα για κάθε εξάμηνο, για να ολοκληρωθεί η διαδικασία.

Σε συνεργασία με τον υπεύθυνο εκπαιδευτικό Πληροφορικής δημιουργήθηκε ένα σενάριο μαθήματος (Παράρτημα Ι). Σύμφωνα με αυτό, πραγματοποιήθηκε εισαγωγικό μάθημα στη γλώσσα προγραμματισμού Java και επεξηγήθηκαν εκ νέου οι βασικές έννοιες των μεταβλητών, της απόδοσης τιμών, των πράξεων (αριθμητικών και λογικών), των κλήσεων συναρτήσεων κτλ, υπό το πρίσμα της Java. Ιδιαίτερο βάρος δόθηκε στις έννοιες των αντικειμένων της καθημερινότητάς τους, που ήταν και το κύριο ζητούμενο του μαθήματος και στη συνέχεια, αυτές συνδέθηκαν παράλληλα με την ανάλυση, που ακολούθησε για το περιβάλλον του Greenfoot. Έγιναν οι απαραίτητες δοκιμές με την τεχνική trial and error, ώστε να κατανοήσουν οι εκπαιδευόμενοι πως λειτουργεί και πως αντιδρά το περιβάλλον, τόσο στη σύνθεση των αντικειμένων, όσο και στην αντιμετώπιση προβλημάτων π.χ. συντακτικά λάθη.

Στη συνέχεια, οι εκπαιδευόμενοι έκαναν δοκιμές με έτοιμα παραδείγματα – παιχνίδια, κάνοντας μικρές παρεμβάσεις στα στιγμιότυπα των κλάσεων επάνω στη σκηνή, εστιάζοντας περισσότερο στις μεθόδους και τα πλεονεκτήματα της κληρονομικότητας. Τέλος, τους δόθηκαν αναλυτικές οδηγίες για τη σύνταξη και τα αποτελέσματα συγκεκριμένων μεθόδων, που κληρονομούνται από την κλάση Actor και τις οποίες χρειάζονται, ώστε να μετασχηματίσουν και να εξελίξουν ένα έτοιμο παιχνίδι.

Στην τελευταία διδακτική ώρα του 8^{ου} μαθήματος, διανεμήθηκε στους εκπαιδευόμενους το ερωτηματολόγιο, συνοδευόμενο από μία κατατοπιστική και διευκρινιστική επιστολή. Αφού αναγνώστηκαν οι διευκρινίσεις, οι συμμετέχοντες κλήθηκαν να απαντήσουν στις ερωτήσεις του ερωτηματολογίου (Παράρτημα ΙΙ).

ΕΝΟΤΗΤΑ 3. ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΕΥΝΑΣ

3.1. Αντικειμενικός σκοπός έρευνας

Ο αντικειμενικός σκοπός της έρευνας είναι να διαπιστωθεί η συμβολή της χρήσης παιχνιδιών κατά την εκπαιδευτική διαδικασία στην κατανόηση και εκμάθηση βασικών όρων και εννοιών του προγραμματισμού. Για το λόγο αυτό, επιλέχθηκαν εκπαιδευόμενοι, οι οποίοι είχαν ήδη διδαχθεί προγραμματισμό με τον παραδοσιακό τρόπο διδασκαλίας, ώστε να είναι σε θέση να συγκρίνουν με βάση και την πρότερη εμπειρία τους.

Επίσης, εξίσου βασικός στόχος είναι να διευκρινιστεί αν και κατά πόσο συμβάλλει η χρήση της συγκεκριμένης εκπαιδευτικής πλατφόρμας, δηλαδή του Greenfoot, ώστε να είναι η προγραμματιστική διαδικασία προσιτή και ευχάριστη στους εκπαιδευόμενους. Είναι σημαντικό να καταγραφούν οι απόψεις τους για τη λειτουργία και τη χρήση του εν λόγω περιβάλλοντος.

3.2. Ερευνητική μέθοδος

Κατά την εκπόνηση της παρούσας εργασίας διενεργήθηκε ποσοτική περιγραφική έρευνα με εκπαιδευτική παρέμβαση. Με τη βοήθεια ερωτηματολογίου συγκεντρώθηκαν οι στάσεις και απόψεις των συμμετεχόντων αναφορικά με το αντικείμενο, το οποίο είχαν διδαχθεί. Κατά συνέπεια, στόχος ήταν η συλλογή πληροφοριών από ένα δείγμα πληθυσμού και η στατιστική ανάλυσή τους, ώστε να διεξαχθούν αξιόπιστα και έγκυρα συμπεράσματα, τα οποία θα δύνανται να οδηγήσουν σε γενίκευση.

Ακολουθώντας τα βασικά στάδια μίας ποσοτικής έρευνας (Cohen et al., 2011) προηγήθηκε όλων των ενεργειών η βιβλιογραφική ανασκόπηση. Στη συνέχεια, διατυπώθηκαν οι στόχοι και οι υποθέσεις της έρευνας και αναζητήθηκαν τα εργαλεία εκείνα, τα οποία θα συνέβαλαν καθοριστικά στον έλεγχο των υποθέσεων (ερωτηματολόγιο). Τέλος, πραγματοποιήθηκε η διεξαγωγή της έρευνας και η στατιστική ανάλυση και παρουσίαση των αποτελεσμάτων της.

3.3. Υποθέσεις έρευνας

Με βάση τους σκοπούς, που αναφέρθηκαν προηγουμένως, διατυπώθηκαν για την παρούσα έρευνα οι ακόλουθες υποθέσεις:

A. Η χρήση παιχνιδιών κατά την εκπαιδευτική διαδικασία συμβάλλει σημαντικά στην κατανόηση βασικών όρων και εννοιών του προγραμματισμού.

B. Η εκπαιδευτική πλατφόρμα Greenfoot αποτελεί ένα προσιτό και ευχάριστο περιβάλλον για την εκμάθηση του προγραμματισμού.

Γ. Οι εκπαιδευόμενοι προτιμούν την εκπαιδευτική πλατφόρμα Greenfoot από ένα συμβατικό περιβάλλον ανάπτυξης μιας γλώσσας προγραμματισμού.

3.4. Δείγμα έρευνας

Όπως έχει ήδη αναφερθεί, στην παρούσα έρευνα συμμετείχαν 107 ενήλικες εκπαιδευόμενοι, άνδρες και γυναίκες, του 1^{ου} Δημόσιου Ινστιτούτου Επαγγελματικής Κατάρτισης (ΔΙΕΚ) Λάρισας. Η συγκεκριμένη σχολική μονάδα επιλέχθηκε για τους εξής λόγους:

- Το Greenfoot είναι μία πλατφόρμα, που απευθύνεται σε μεγαλύτερης ηλικίας εκπαιδευόμενους (άνω των 14 ετών), οπότε θα έπρεπε να αναζητηθεί δείγμα, είτε στην τελευταία τάξη του γυμνασίου, είτε σε λυκειακές τάξεις. Ωστόσο, δεν βρέθηκε κάποιο τμήμα, στο οποίο να διδάσκεται το Greenfoot (Φυσικά, δεν ελέγχθηκαν όλα τα σχολεία και τμήματα της πόλης), κάτι το οποίο επρόκειτο να γίνει στο εν λόγω ΔΙΕΚ.
- Στα ΔΙΕΚ οι διδακτικές ώρες των μαθημάτων είναι συνεχόμενες (δηλαδή ουσιαστικά κάθε μάθημα δεν ολοκληρώνεται σε μία διδακτική ώρα (45 λεπτά), αλλά σε δύο ή και περισσότερες), γεγονός ιδιαίτερα βοηθητικό, καθώς τόσο ο εκπαιδευτικός όσο και οι εκπαιδευόμενοι έχουν τον απαραίτητο χρόνο διεξοδικής διδασκαλίας και ουσιαστικής εμπέδωσης των γνώσεων αντίστοιχα.
- Η ερευνήτρια γνωρίζει επί προσωπικού τον καθηγητή Πληροφορικής, που δίδασκε το αντικείμενο, με συνέπεια να είναι εύκολη και αποτελεσματική τόσο η διδασκαλία, όσο και η διεξαγωγή της έρευνας. Άλλωστε, ο ίδιος εφάρμοσε στην πράξη και υπό τις οδηγίες της ερευνήτριας το σενάριο μαθήματος, που του δόθηκε.
- Για την έρευνα δεν απαιτήθηκαν ειδικές άδειες συμμετοχής καθώς πρόκειται για ενήλικες εκπαιδευόμενους, οι οποίοι, ωστόσο, ρωτήθηκαν εάν επιθυμούν να συμμετάσχουν και δέχτηκαν όλοι στο σύνολό τους. Επίσης, ζητήθηκε άδεια από τον διευθυντή του εν λόγω εκπαιδευτικού ιδρύματος, ο οποίος συμφώνησε, εφόσον υπήρχε η συναίνεση των εκπαιδευόμενων.
- Η πόλη της Λάρισας είναι ο τόπος μόνιμης κατοικίας της ερευνήτριας, επομένως ήταν δυνατή η άμεση επαφή, η διευκρίνιση οποιωνδήποτε αποριών και η επίλυση πιθανών προβλημάτων.

3.5. Εργαλείο έρευνας

Για τη διεξαγωγή της παρούσας έρευνας χρησιμοποιήθηκε ως εργαλείο το ερωτηματολόγιο (Παράρτημα II). Συντάχθηκε ένα ερωτηματολόγιο 15 ερωτήσεων κλειστού τύπου (δύο από αυτές έδιναν τη δυνατότητα συμπλήρωσης στοιχείου), το οποίο περιελάμβανε επιπλέον δύο στοιχεία δημογραφικού χαρακτήρα, το φύλο και την ηλικία των συμμετεχόντων. Δεν κρίθηκε απαραίτητο να ζητηθούν άλλα δημογραφικά στοιχεία, όπως π.χ. επίπεδο μόρφωσης, διότι όλοι οι συμμετέχοντες ήταν υποχρεωτικά απόφοιτοι δευτεροβάθμιας εκπαίδευσης. Το ερωτηματολόγιο μοιράστηκε στα τμήματα, προκειμένου να συμπληρωθεί από τους συμμετέχοντες μέσα στην τάξη.

Στο ερωτηματολόγιο είχε επισυναφθεί μία επιστολή, όπου γινόταν αναφορά στην ερευνήτρια, στο εκπαιδευτικό ίδρυμα, στο οποίο φοιτούσε και στον βασικό σκοπό και στόχο της έρευνας. Παράλληλα, δόθηκαν ορισμένες διευκρινίσεις για τον τρόπο συμπλήρωσης του ερωτηματολογίου και επισημάνθηκε η διατήρηση της ανωνυμίας των συμμετεχόντων και η χρήση των αποτελεσμάτων αποκλειστικά και μόνο για τη στατιστική ανάλυση της έρευνας.

Το ερωτηματολόγιο, στην αρχή του οποίου εμφανίζονται τα δημογραφικά στοιχεία (φύλο, ηλικία), χωρίστηκε σε τρεις ενότητες: α) Ερωτήσεις για την πρότερη εμπειρία των εκπαιδευόμενων στη χρήση Η/Υ, τον προγραμματισμό και άλλα περιβάλλοντα, β) Ερωτήσεις που δείχνουν κατά πόσο έγινε κατανοητό το διδακτικό αντικείμενο και γ) Ερωτήσεις σχετικά με το περιβάλλον του Greenfoot και την άποψη, που σχημάτισαν γι' αυτό οι συμμετέχοντες.

Η επιλογή του ερωτηματολογίου ήταν η ενδεδειγμένη για τη διεξαγωγή της έρευνας. Πρώτα από όλα, διανεμήθηκε στους εκπαιδευόμενους/συμμετέχοντες μέσα στην τάξη, ώστε να διασφαλιστεί ότι θα παραληφθούν συμπληρωμένα στο σύνολό τους. Επίσης, επρόκειτο για μία άμεση και μικρής διάρκειας διαδικασία, που δεν θα κούραζε τους συμμετέχοντες. Τέλος, το ερωτηματολόγιο με τις κλειστού τύπου ερωτήσεις διευκόλυνε τη συλλογή και επεξεργασία των δεδομένων.

3.6. Ανάλυση αποτελεσμάτων

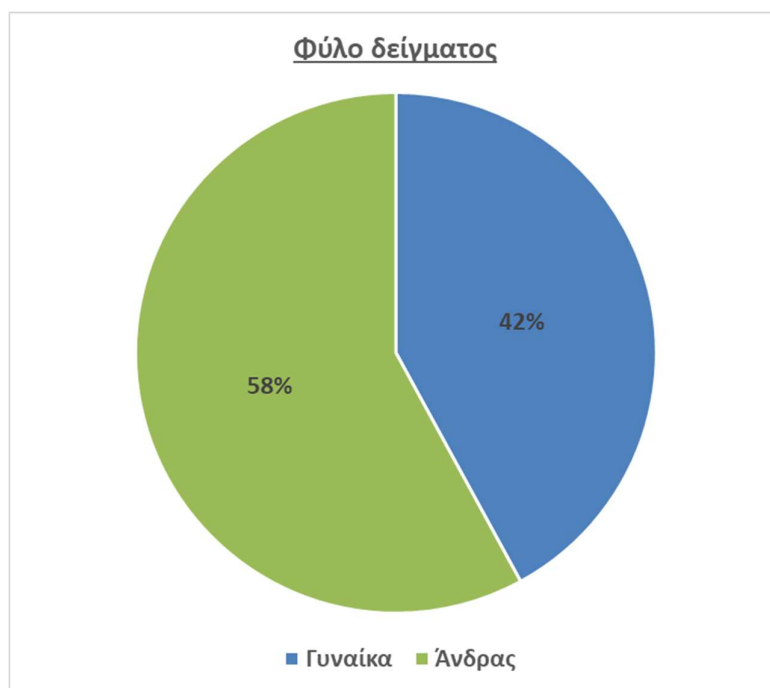
Μετά τη συλλογή των ερωτηματολογίων τα στοιχεία, που αυτά περιελάμβαναν, κωδικοποιήθηκαν και αποθηκεύτηκαν σε φύλλα εργασίας του λογιστικού λογισμικού Microsoft Excel. Για την αποτύπωση των αποτελεσμάτων και τη δημιουργία συσχετίσεων χρησιμοποιήθηκαν τα είδη γραφήματος πίτα και ράβδοι, που διαθέτει το εν λόγω πρόγραμμα. Στην επόμενη ενότητα ακολουθεί η περιγραφή και ανάλυση των αποτελεσμάτων της έρευνας.

ΕΝΟΤΗΤΑ 4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΕΥΝΑΣ

4.1. Δημογραφικά στοιχεία

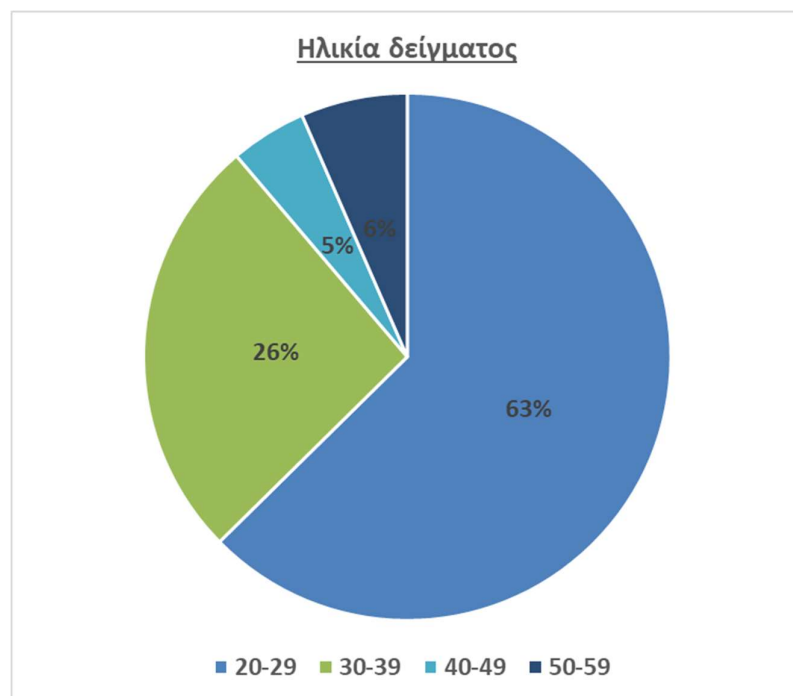
Πέρα από τις τρεις ομάδες ερωτήσεων, το ερωτηματολόγιο περιελάμβανε και δύο στοιχεία δημογραφικού χαρακτήρα. Αυτά είναι η ηλικία και το φύλο.

Ως προς το φύλο (Πίνακας 2), από τους 107 συμμετέχοντες οι 45 ήταν γυναίκες (ποσοστό 42%) και οι άνδρες ήταν 62 (ποσοστό 58%). Το δείγμα θεωρείται αντικειμενικό καθώς πιστεύεται ότι τα δύο φύλα εκπροσωπήθηκαν σχεδόν ισόποσα.



Πίνακας 2. Φύλο δείγματος

Ως προς την ηλικία (Πίνακας 3), το 63% των συμμετεχόντων ήταν μεταξύ 20 και 29 ετών (67 άτομα), το 26% ήταν μεταξύ 30 και 39 ετών (28 άτομα), ενώ υπήρχαν 5 άτομα μεταξύ 40 και 49 ετών και 7 άτομα (ποσοστό 6%) σε ένα ηλικιακό εύρος από 50 έως 59 έτη.



Πίνακας 3. Ηλικία δείγματος

4.2. Στοιχεία γενικών γνώσεων

Η πρώτη ενότητα του ερωτηματολογίου περιλαμβάνει πέντε ερωτήσεις, οι οποίες αφορούν στις γενικές γνώσεις χρήσης Η/Υ και προγραμματισμού των συμμετεχόντων. Επιλέχθηκαν, προκειμένου να σχηματιστεί μία εικόνα σχετικά με το υπόβαθρο των ερωτηθέντων, ώστε να αξιολογηθούν σφαιρικά τα αποτελέσματα της έρευνας.

Στην πρώτη ερώτηση αυτής ενότητας (Πίνακας 4), οι συμμετέχοντες κλήθηκαν να αξιολογήσουν το επίπεδο γνώσεών τους ως προς τη χρήση Η/Υ σε μία κλίμακα από το 1 (Λίγες γνώσεις) ως το 5 (Άριστες). Το 61% χαρακτήρισε άριστο το επίπεδο γνώσεων και μαζί με το 19% που απάντησε πως βρίσκεται σε πολύ καλό επίπεδο αποτελούν ένα ιδιαίτερα ικανοποιητικό ποσοστό για τα δεδομένα της έρευνας. Το 14% των συμμετεχόντων δήλωσε μέτριο επίπεδο στη χρήση Η/Υ, το 8% καλό και το 5% φαίνεται να κατέχει λίγες γνώσεις.



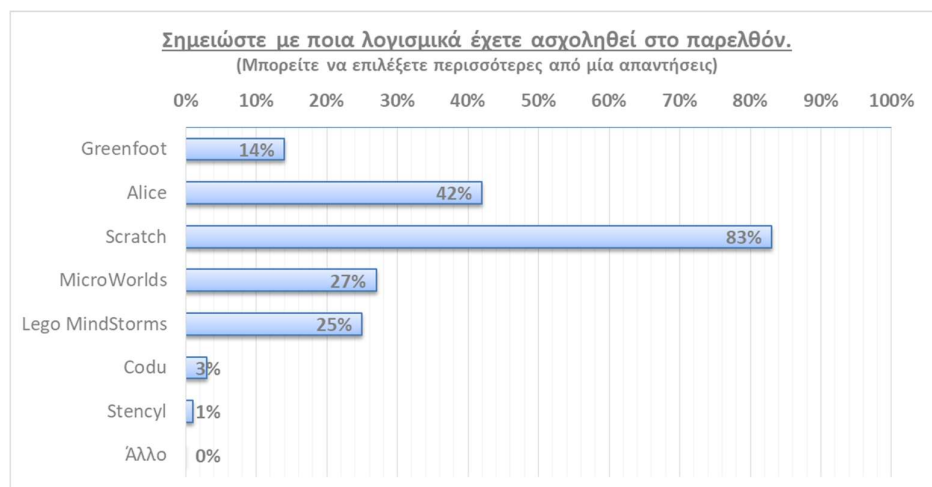
Πίνακας 4. Γνώσεις Η/Υ

Σε επόμενη ερώτηση, το 46% των εκπαιδευόμενων δήλωσε ότι έχει άριστες γνώσεις προγραμματισμού σε αντιδιαστολή με το 11% αυτών, που έχει λίγες γνώσεις προγραμματισμού. Ποσοστό της τάξης του 18% έχει πολύ καλές και καλές γνώσεις, ενώ το 14% χαρακτηρίζει μέτριο το επίπεδο των γνώσεών του ως προς τον προγραμματισμό.



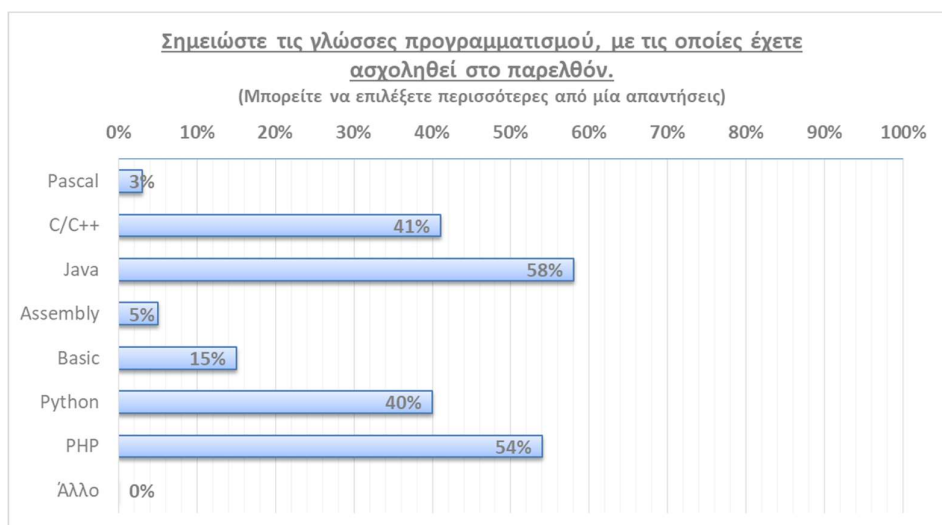
Πίνακας 5. Γνώσεις σε προγραμματισμό

Σημαντικό θεωρήθηκε να συμπεριληφθεί στο ερωτηματολόγιο μία ερώτηση, που θα έδειχνε τα λογισμικά, με τα οποία έχουν ασχοληθεί οι συμμετέχοντες κατά το παρελθόν (Πίνακας 6). Κατά συντριπτική πλειοψηφία οι εκπαιδευόμενοι είχαν ασχοληθεί με το περιβάλλον Scratch, ενώ και το 42% αυτών είχε έρθει σε επαφή με το εκπαιδευτικό περιβάλλον Alice. Αξίζει να σημειωθεί ότι μόλις το 14% είχε ασχοληθεί με το Greenfoot ξανά στο παρελθόν.



Πίνακας 6. Λογισμικά (πρότερη γνώση)

Ιδιαίτερη σημασία είχε για την έρευνα και η διαπίστωση πρότερης γνώσης των εκπαιδευόμενων, όσον αφορά στις γλώσσες προγραμματισμού. Συγκεκριμένα, ας αναφερθεί ότι το 58% των συμμετεχόντων είχε ασχοληθεί στο παρελθόν με τη γλώσσα προγραμματισμού Java. Το στοιχείο αυτό ήταν σημαντικό για τα συμπεράσματα της έρευνας, καθώς οι συμμετέχοντες θα ήταν σε θέση να συγκρίνουν τη διδασκαλία του ίδιου αντικειμένου με διαφορετικό τρόπο (Πίνακας 7).



Πίνακας 7. Γλώσσες προγραμματισμού (πρότερη γνώση)

Η τελευταία ερώτηση αυτής της ενότητας του ερωτηματολογίου αφορά στη χρήση γλώσσας προγραμματισμού από τους συμμετέχοντες στην εργασία τους, ποσοστό που άγγιζε μόλις το 17%.

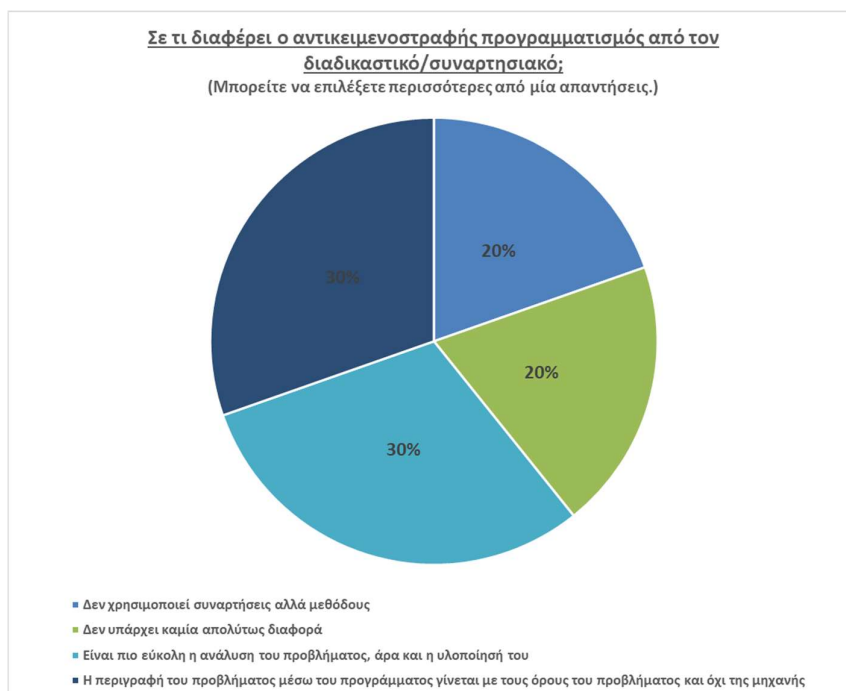


Πίνακας 8. Χρήση γλώσσας προγραμματισμού στην εργασία

4.3. Στοιχεία κατανόησης αντικειμένου

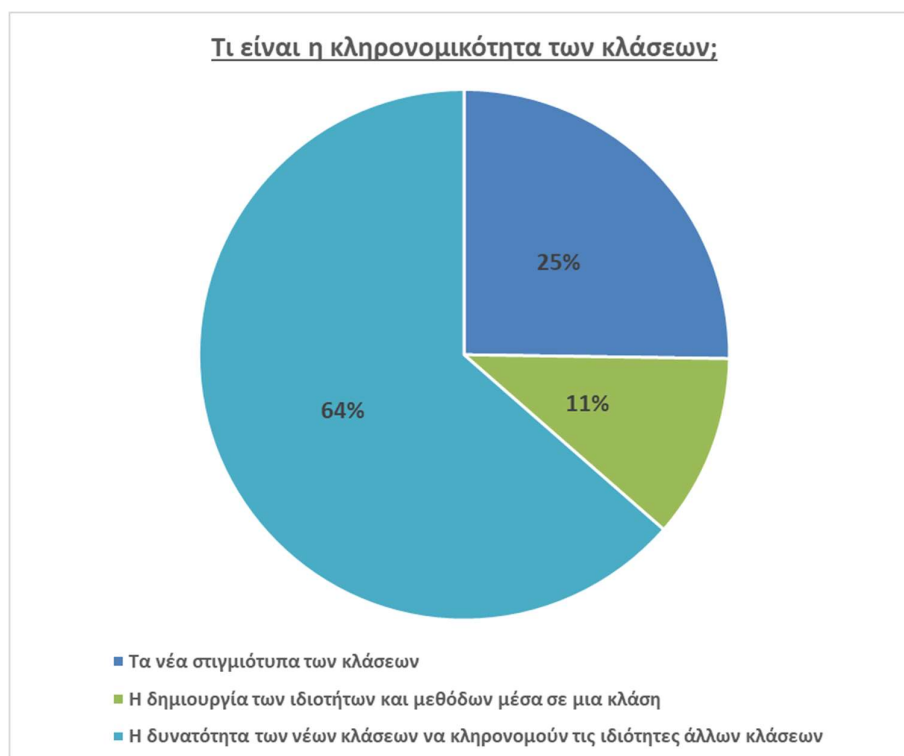
Η δεύτερη ενότητα του ερωτηματολογίου περιλαμβάνει πέντε κλειστού τύπου ερωτήσεις, οι οποίες αφορούν στην κατανόηση του αντικειμένου και αφορούν θέματα, τα οποία φυσικά διδάχθηκαν οι εκπαιδευόμενοι. Οι ερωτήσεις περιλαμβάνουν βασικές έννοιες και λειτουργίες του περιβάλλοντος ανάπτυξης και δύνανται να δώσουν μία σφαιρική εικόνα, για τις γνώσεις, που αποκόμισαν οι συμμετέχοντες.

Στην πρώτη ερώτηση αυτής της ενότητας, οι συμμετέχοντες κλήθηκαν απαντήσουν για τις διαφορές αντικειμενοστραφούς και διαδικαστικού/συναρτησιακού προγραμματισμού. Οι σωστές απαντήσεις είναι δύο: α) Η περιγραφή του προβλήματος μέσω του προγράμματος γίνεται με τους όρους του προβλήματος και όχι της μηχανής και β) Είναι πιο εύκολη η ανάλυση του προβλήματος, άρα και η υλοποίησή του. Συνολικά, το 60% των απαντήσεων ήταν σωστές (Πίνακας 9).



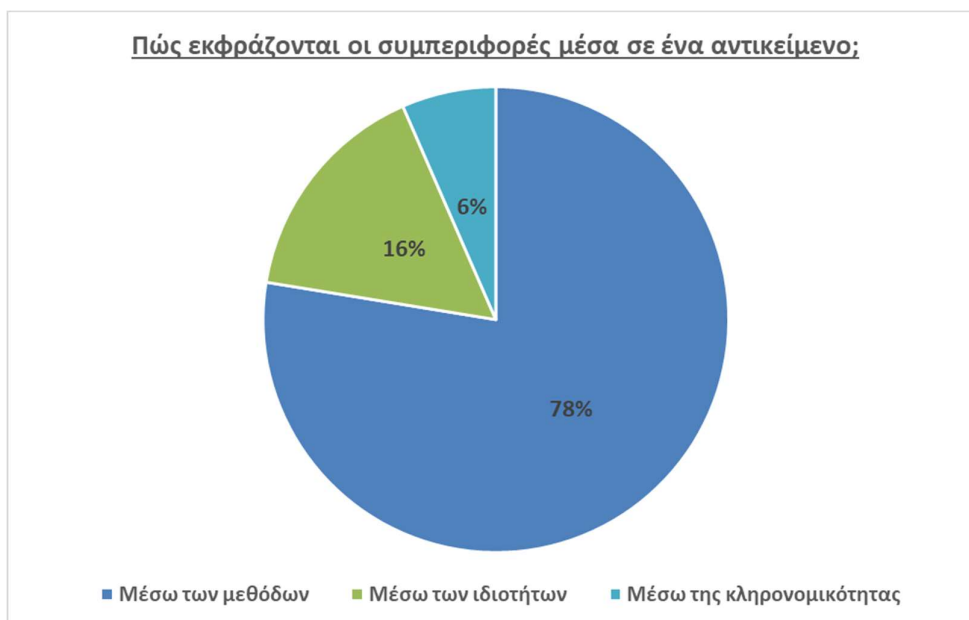
Πίνακας 9. Διαφορές αντικειμενοστραφούς και διαδικαστικού/συναρτησιακού προγραμματισμού

Στην ερώτηση «Τι είναι η κληρονομικότητα των κλάσεων;» σωστή είναι η απάντηση *Η δυνατότητα των κλάσεων να κληρονομήσουν τις ιδιότητες άλλων*, την οποία και επέλεξε το 64% των εκπαιδευόμενων. Το υπόλοιπο 36% απάντησε λανθασμένα.



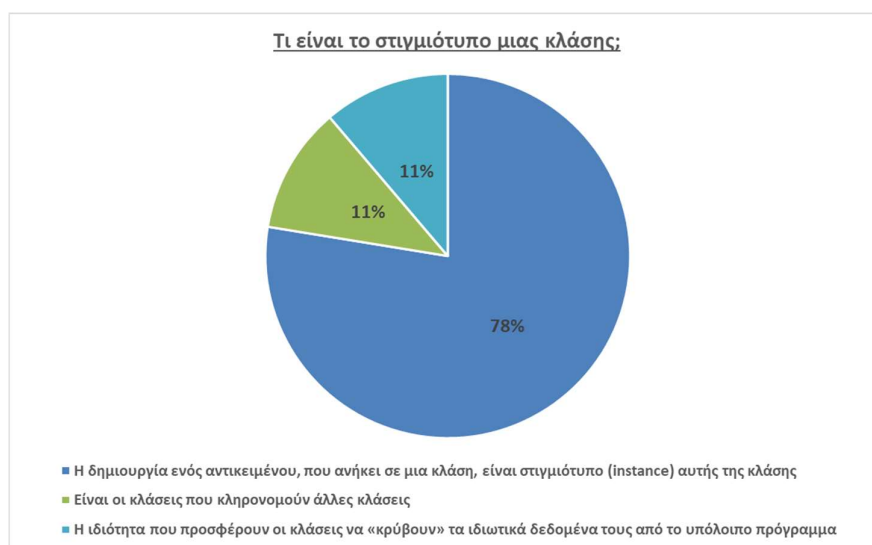
Πίνακας 10. Κληρονομικότητα κλάσεων

Στην επόμενη ερώτηση το 78% των ερωτηθέντων απάντησε σωστά και επέλεξε *Μέσω των μεθόδων*, ενώ το 22% επέλεξε λανθασμένες απαντήσεις (Πίνακας 11).



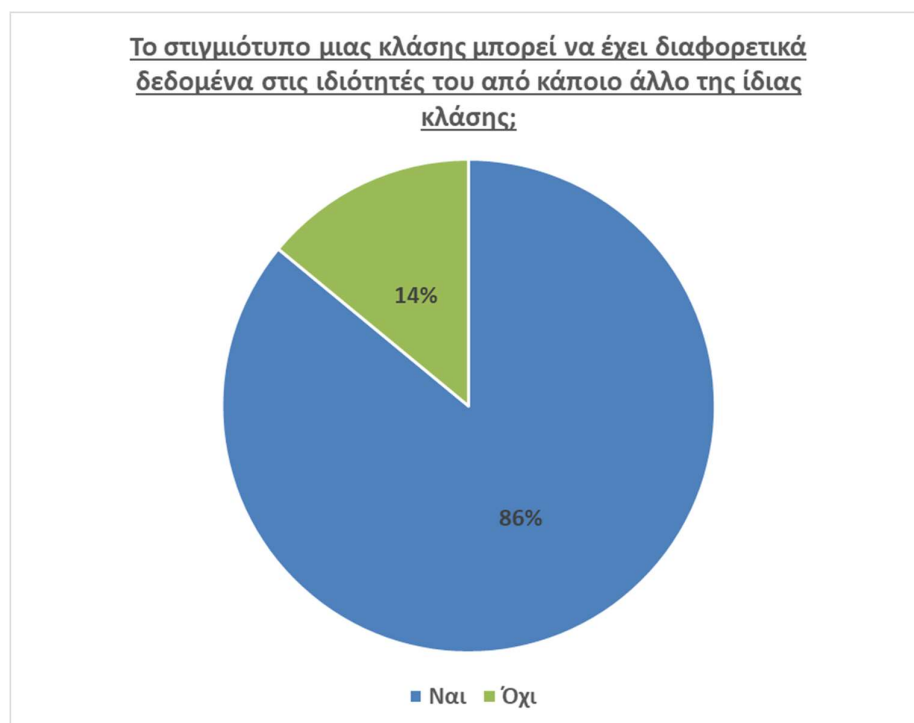
Πίνακας 11. Τρόπος έκφρασης συμπεριφορών μέσα σε αντικείμενο

Αντίστοιχα ποσοστά σημειώθηκαν και σε αυτήν την ερώτηση (Πίνακας 12) με το 78% των ερωτηθέντων να απαντά σωστά, ενώ οι λανθασμένες απαντήσεις έλαβαν έκαστη από ένα ποσοστό της τάξης του 11%.



Πίνακας 12. Στιγμιότυπο κλάσης

Στην τελευταία ερώτηση αυτής της ενότητας η σωστή απάντηση *Ναι* επιλέχθηκε από το 86% των εκπαιδευόμενων με μόλις το 14% να δίνει λανθασμένη απάντηση (Πίνακας 13).

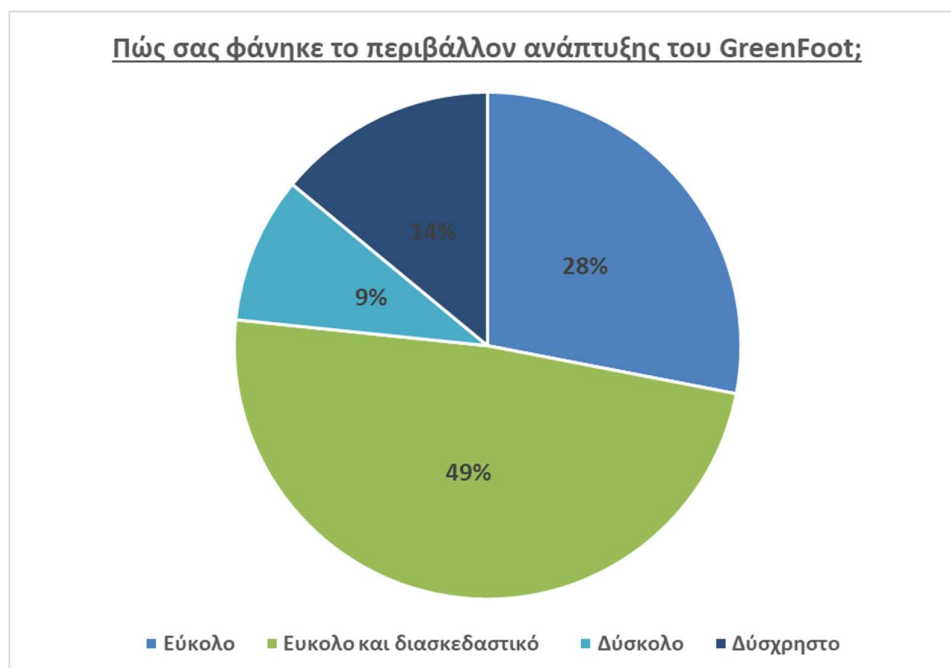


Πίνακας 13. Ιδιότητες στιγμιότυπου κλάσης

4.4. Στοιχεία περιβάλλοντος

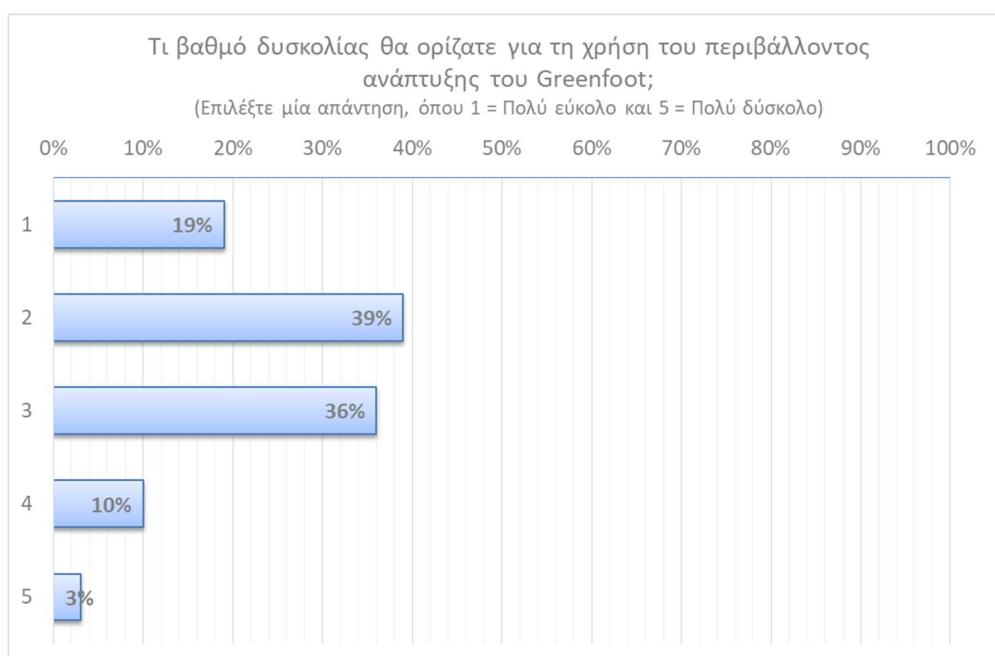
Η τρίτη ενότητα του ερωτηματολογίου περιλαμβάνει πέντε κλειστού τύπου ερωτήσεις, οι οποίες αφορούν στο περιβάλλον του Greenfoot και την εντύπωση, που αποκόμισαν γι' αυτό οι συμμετέχοντες στην έρευνα.

Στην ερώτηση «Πώς σας φάνηκε το περιβάλλον ανάπτυξης του Greenfoot», το 49% των ερωτηθέντων δήλωσε ότι το περιβάλλον είναι *Εύκολο και διασκεδαστικό* και μαζί με το ποσοστό που θεωρεί ότι το περιβάλλον είναι *Εύκολο*, αποτελούν την πλειοψηφία και αγγίζουν το ποσοστό του 77%. Το 14% των εκπαιδευόμενων προφανώς αντιμετώπισε δυσκολίες, καθώς δήλωσε ότι το βρήκε *Δύσχερστο*, ενώ και ένα ποσοστό της τάξης του 9% θεωρεί το περιβάλλον *Δύσκολο* (Πίνακας 14).



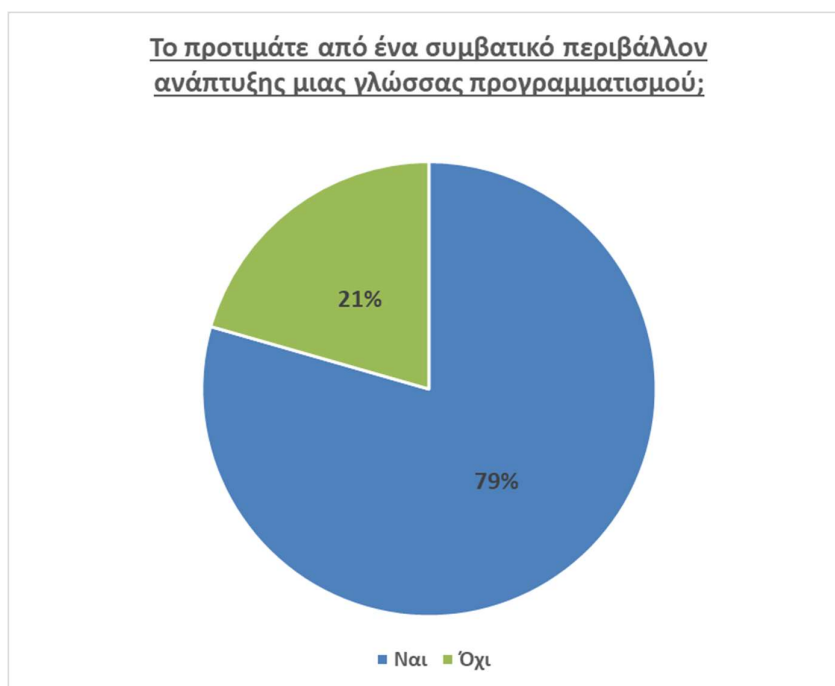
Πίνακας 14. Αποψη για το περιβάλλον Greenfoot

Στην επόμενη ερώτηση (Πίνακας 15), που αφορά στο βαθμό δυσκολίας ως προς τη χρήση του περιβάλλοντος σε μία κλίμακα από το 1 ως το 5, όπου 1=Πολύ εύκολο και 5=Πολύ δύσκολο, το 19% απάντησε *Πολύ εύκολο*, το 39% απάντησε *Εύκολο*, το 36% βρήκε το περιβάλλον *Μέτριας δυσκολίας*, το 10% απάντησε *Δύσκολο* και το 3% *Πολύ δύσκολο*.



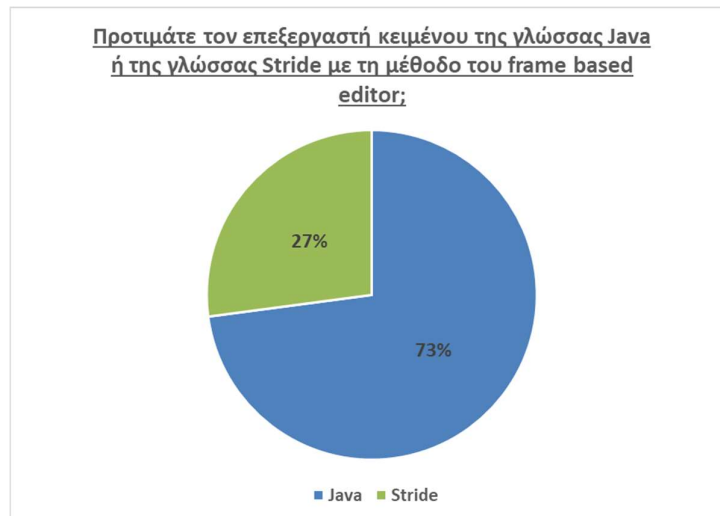
Πίνακας 15. Βαθμός δυσκολίας

Ιδιαίτερα αυξημένο ήταν το ποσοστό των εκπαιδευόμενων, που δήλωσε ότι προτιμάει το περιβάλλον ανάπτυξης του Greenfoot από ένα συμβατικό περιβάλλον μιας γλώσσας προγραμματισμού, όπου απαιτείται αποκλειστικά η χρήση κώδικα. Μόλις το 21% επέλεξε ένα συμβατικό περιβάλλον, ενώ το 79% επέλεξε το περιβάλλον του *Greenfoot*.



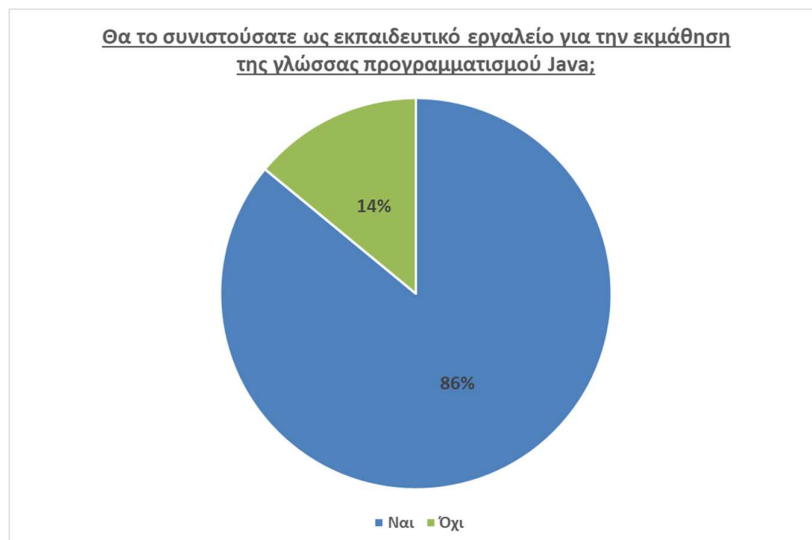
Πίνακας 16. *Greenfoot* ή συμβατικό περιβάλλον προγραμματισμού;

Στο ίδιο σκεπτικό κινήθηκε και η επόμενη ερώτηση (Πίνακας 17), όπου κλήθηκαν οι ερωτηθέντες να επιλέξουν μεταξύ του επεξεργαστή κειμένου της Java και της Stride. Το 73% απάντησε ότι προτιμά να εργάζεται με τον επεξεργαστή της γλώσσας *Java* και το υπόλοιπο 27% προτίμησε τη *Stride*.



Πίνακας 17. Επεξεργαστής κειμένου Java ή Stride;

Στην τελευταία ερώτηση, οι εκπαιδευόμενοι ρωτήθηκαν για το αν θα συνιστούσαν το Greenfoot ως εκπαιδευτικό εργαλείο εκμάθησης της Java. Μόλις το 14% αυτών απάντησε *Όχι*, ενώ η συντριπτική πλειοψηφία του 86% απάντησε *Ναι*, δηλαδή ότι θα το πρότεινε (Πίνακας 18).



Πίνακας 18. Σύσταση Greenfoot ως εκπαιδευτικό εργαλείο εκμάθησης Java

ΕΝΟΤΗΤΑ 5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΕΥΝΑΣ

Ολοκληρώνοντας την περιγραφή και ανάλυση των αποτελεσμάτων της έρευνας, αντιλαμβάνεται κανείς ότι οι στόχοι, που τέθηκαν εξ αρχής, έχουν επιτευχθεί σε αρκετά ικανοποιητικό βαθμό.

Ειδικότερα, από τις απαντήσεις στις ερωτήσεις κατανόησης του διδακτικού αντικειμένου και το μεγάλο ποσοστό σωστών απαντήσεων διαφαίνεται ότι οι εκπαιδευόμενοι κατανόησαν και εμπέδωσαν τις βασικές έννοιες και όρους του αντικειμενοστραφούς προγραμματισμού, που διδάχθηκαν. Παράλληλα, ιδιαίτερα υψηλό ήταν το ποσοστό των εκπαιδευόμενων, που χαρακτήρισαν την πλατφόρμα Greenfoot ως περιβάλλον «Εύκολο στη χρήση», προκειμένου κανείς να διδαχθεί προγραμματισμό. Τέλος, κατά πλειοψηφία (ποσοστό 79%) οι εκπαιδευόμενοι επιλέγουν το περιβάλλον Greenfoot από ένα συμβατικό περιβάλλον ανάπτυξης μιας γλώσσας προγραμματισμού.

Επιπροσθέτως, κατά ομολογία του υπεύθυνου εκπαιδευτικού Πληροφορικής, κ. Γεωργίου Παπαϊωάννου, η εκπαιδευτική διαδικασία διατήρησε αμείωτο το ενδιαφέρον των εκπαιδευόμενων, οι οποίοι συμμετείχαν ενεργά όχι μόνο κατά το μετασχηματισμό του παιχνιδιού, αλλά ακόμα και στο θεωρητικό κομμάτι της διδασκαλίας, εκφράζοντας απορίες και κρατώντας αναλυτικές σημειώσεις. Ιδιαίτερα ενθαρρυντικό ήταν το γεγονός ότι οι εκπαιδευόμενοι, μετά τις πρώτες οδηγίες από τον εκπαιδευτικό τους, κατόρθωσαν να λάβουν τις αποφάσεις εκείνες και να εφαρμόσουν τις πρακτικές εκείνες, που τους οδήγησαν στην εξέλιξη και ολοκλήρωση του παιχνιδιού χωρίς περαιτέρω καθοδήγηση από τον διδάσκοντα. Τέλος, ακόμα και μετά την ολοκλήρωση του συγκεκριμένου πειράματος, οι εκπαιδευόμενοι θέλησαν να συνεχίσουν τη διδασκαλία και με το τέλος του εξαμήνου εξέλιξαν ακόμη περισσότερο το παιχνίδι, που είχαν ξεκινήσει.

Διαφαίνεται, λοιπόν, με βάση τις απαντήσεις των συμμετεχόντων, ότι ένα οπτικό περιβάλλον προγραμματισμού, όπως είναι το Greenfoot, συμβάλλει σημαντικά στη βελτίωση της εκπαιδευτικής διαδικασίας καθώς ενεργοποιεί τους εκπαιδευόμενους, ενισχύει το ενδιαφέρον τους για το αντικείμενο και τονώνει την αυτοπεποίθησή τους, καθώς αντιλαμβάνονται και εμπεδώνουν τις έννοιες και τους όρους, που απαντούν. Η εξοικείωση με το φιλικό και εύχρηστο περιβάλλον του εκπαιδευτικού εργαλείου Greenfoot βοήθησε τους συμμετέχοντες να ξεπεράσουν φόβους και ενδοιασμούς, με αποτέλεσμα να χρησιμοποιήσουν απρόσκοπτα τις ικανότητές τους, ώστε να ολοκληρωθεί επιτυχώς ο μετασχηματισμός του παιχνιδιού.

Θέτοντας ως βάση το ποσοστό των εκπαιδευόμενων, που θα συνέστηνε το οπτικό περιβάλλον προγραμματισμού, Greenfoot, για την εκμάθηση της γλώσσας προγραμματισμού Java (86% των συμμετεχόντων, δηλαδή 92 από αυτούς σε σύνολο 107 ατόμων), είναι σαφές ότι το συγκεκριμένο εκπαιδευτικό εργαλείο δύναται να συνδράμει αποτελεσματικά στην εκμάθηση του αντικειμενοστραφούς προγραμματισμού. Είναι ένα περιβάλλον, το οποίο οδηγεί σε μεγάλο βαθμό σε

εξοικείωση, κατανόηση, εμπέδωση και τελικά εκμάθηση δύσκολων και περίπλοκων όρων και εννοιών, χωρίς να απαιτείται κατ' ανάγκη η συγγραφή κώδικα.

Αξίζει, επίσης, να αναφερθεί ότι η επιτυχία της χρήσης παιχνιδιών κατά τη διδασκαλία βασικών εννοιών προγραμματισμού αντανακλάται και στην απάντηση των εκπαιδευόμενων ως προς την προτίμηση του περιβάλλοντος Greenfoot σε σχέση με ένα συμβατικό περιβάλλον ανάπτυξης γλώσσας προγραμματισμού. Από τους 107 συμμετέχοντες οι 85 απάντησαν ότι προτιμούν το Greenfoot. Το ποσοστό αυτό είναι ιδιαίτερης σημασίας καθώς οι ερωτηθέντες σε υψηλό ποσοστό είχαν ασχοληθεί με τον προγραμματισμό στο παρελθόν, αλλά με διαφορετικό τρόπο διδασκαλίας. Είναι φανερό ότι λόγω αυτής της πρότερης εμπειρίας τους είναι σε θέση να κρίνουν και να συγκρίνουν αντικειμενικά τον παραδοσιακό τρόπο διδασκαλίας του προγραμματισμού και τις νέες τεχνικές μάθησης, όπως είναι και η χρήση παιχνιδιών.

ΕΝΟΤΗΤΑ 6. ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΓΑΣΙΑΣ

Ο αντικειμενικός σκοπός της παρούσας εργασίας ήταν να διαπιστωθεί η συμβολή της χρήσης παιχνιδιών κατά την εκπαιδευτική διαδικασία στην κατανόηση και εκμάθηση βασικών όρων και εννοιών του προγραμματισμού. Επίσης, ήταν σημαντικό να διερευνηθεί η συμβολή συγκεκριμένης εκπαιδευτικής πλατφόρμας, δηλαδή του Greenfoot, ώστε να είναι η προγραμματιστική διαδικασία προσιτή και ευχάριστη στους εκπαιδευόμενους.

Με βάση τα συμπεράσματα της έρευνας, όπως αυτά αναλύθηκαν στην προηγούμενη ενότητα, εύκολα διαπιστώνει κανείς ότι τα ψηφιακά παιχνίδια αποτελούν μέσο μάθησης, που έχει ως απώτερο σκοπό τη διατήρηση του ενδιαφέροντος των εκπαιδευόμενων και τη μετάδοση της γνώσης με ουσιαστικό και βιωματικό τρόπο. Η δυνατότητα, που παρέχουν, ώστε η επίτευξη των στόχων να είναι άμεσα ορατή, διατηρεί αμείωτο το ενδιαφέρον του εκπαιδευόμενου (Prensky, 2003) και του παρέχει αίσθημα ευφορίας, σιγουριάς και ικανοποίησης, αφού η εκπαιδευτική διαδικασία καθίσταται μια διασκεδαστική δραστηριότητα με γρίφους, στόχους και αποτελέσματα. Άλλωστε, είναι γενικά αποδεκτό ότι τα ψηφιακά παιχνίδια αποτελούν αναπόσπαστο τμήμα της ζωής των νέων και αποτελούν για εκείνους ένα οικείο και ευχάριστο περιβάλλον.

Προς αυτήν την κατεύθυνση κινείται και μία πληθώρα ερευνητών (Dempsey et al., 2002; Rosas et al., 2003; Pivec et al., 2004; Connolly et al., 2008 κ.α.), οι οποίοι υποστηρίζουν ότι τα ψηφιακά παιχνίδια αποτελούν έναν ελκυστικό τρόπο προσέγγισης των νέων, μιας και είναι ιδιαίτερα δημοφιλή σε αυτούς, και αν χρησιμοποιηθούν σωστά κατά τη διδασκαλία, είναι πολύ πιθανό να οδηγήσουν στην επίτευξη των μαθησιακών στόχων. Επιτυγχάνεται, λοιπόν, ένας συνδυασμός μάθησης και ψυχαγωγίας, που μπορεί να οδηγήσει στην αποτελεσματική μάθηση και να συμβάλλει καθοριστικά,

ώστε η εκπαιδευτική διαδικασία να αποκτήσει ένα προσιτό και σύγχρονο πρόσωπο και οι εκπαιδευόμενοι να μελετούν περισσότερο, καθώς αποφασίζουν οι ίδιοι για το είδος γνώσης, που επιθυμούν να λάβουν (Μικρόπουλος, 2006), γεγονός που καθιστά τη μελέτη ουσιαστική, αλλά και ευχάριστη. Ειδικότερα, η δυνατότητα δημιουργίας ψηφιακού παιχνιδιού εξ ολοκλήρου από τον εκπαιδευόμενο οδηγεί τόσο στην επίτευξη των μαθησιακών στόχων, όσο και στην απομάκρυνση τυχόν αρνητικών επιδράσεων από τη χρήση έτοιμων παιχνιδιών.

Παράλληλα, καθοριστική σημασία στην εκπαιδευτική διαδικασία κατέχει και το προγραμματιστικό περιβάλλον, που θα χρησιμοποιηθεί. Τα περιβάλλοντα, που επιλέγονται για τη διδασκαλία σε αρχάριους προγραμματιστές, χρησιμοποιούν ελκυστικά προς τον χρήστη γραφικά και προάγουν έναν απλοποιημένο τρόπο προσέγγισης μεταξύ εφαρμογής και χρήστη/εκπαιδευόμενου. Η χρήση οπτικών στοιχείων συμβάλλει σημαντικά τόσο στην κατανόηση, όσο και στην εφαρμογή βασικών αλγοριθμικών δομών, δίνοντας στη συνέχεια τη δυνατότητα στον εκπαιδευόμενο να «κατακτήσει» και τις παραδοσιακές γλώσσες προγραμματισμού. Σε αυτήν την κατηγορία περιβάλλοντων δείχνει να εντάσσεται και η εκπαιδευτική πλατφόρμα Greenfoot, σύμφωνα με την έρευνα της παρούσας εργασίας, αλλά και με αποτελέσματα άλλων ερευνών, όπως αυτές αποτυπώνονται σε συλλογικό πίνακα προηγούμενης ενότητας (σελ. 30-32).

Η διαμόρφωση των κατάλληλων εκπαιδευτικών προσεγγίσεων (χρήση ψηφιακών παιχνιδιών εν προκειμένω) σε συνδυασμό με τον άρτια καταρτισμένο εκπαιδευτή αναμένεται να φέρει το επιθυμητό αποτέλεσμα και να επιτύχει τον στόχο της εκπαιδευτικής διαδικασίας. Έγκειται στην ευχέρεια του εκπαιδευτικού να βρει τα παιχνίδια εκείνα, που θα καλύψουν όσο το δυνατόν πληρέστερα τις ανάγκες του μαθήματος, θα καταστήσουν τους εκπαιδευόμενους μέρος της μαθησιακής διαδικασίας και θα οδηγήσουν στην απόκτηση της γνώσης. Ειδικότερα, κατά τη διδασκαλία του προγραμματισμού ένας ανάλογος συνδυασμός δύναται να οδηγήσει τον εκπαιδευόμενο στην αποτελεσματική αντιμετώπιση των δυσκολιών, παρανοήσεων και λαθών μέσα από την αποσαφήνιση και κατανόηση προγραμματιστικών εννοιών.

Ολοκληρώνοντας, άποψη της ερευνήτριας είναι ότι η εκπαιδευτική διαδικασία οφείλει να εξελίσσεται, να ακολουθεί την εποχή και τις αλλαγές, που διαδραματίζονται και να συμπορεύεται με τις ανάγκες, που προκύπτουν κάθε φορά. Δεν μπορεί και δεν πρέπει η μάθηση να εγκλωβίζεται σε στερεότυπα και στενά όρια, ούτε και να περιορίζεται σε ό,τι ήδη έχει εφαρμοστεί. Ο εκπαιδευτικός οφείλει να ερευνά, να ενημερώνεται, να εξελίσσεται ο ίδιος και να εξελίξει τη μαθησιακή διαδικασία με σύγχρονες μεθόδους και τεχνικές, που είναι δυνατόν να εφαρμοστούν στην τάξη αυτόνομα ή σε συνδυασμό ακόμα με τον παραδοσιακό τρόπο διδασκαλίας. Αυτός ο τρόπος σκέψης και δράσης, εφόσον λειτουργήσει σωστά, δείχνει να μπορεί να επαναφέρει στον εκπαιδευόμενο την αγάπη για μάθηση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΕΛΛΗΝΟΓΛΩΣΣΗ

- Αλεξοπούλου, Ε., & Κυνηγός, Χ. (2008). *Οι κανόνες μισοψημένων παιχνιδιών ως πλαίσιο κατανόησης και εφαρμογής της δομής επιλογής. Στο Β. Κόμης (επιμ.), 4ο Πανελλήνιο Συνέδριο Διδακτική της Πληροφορικής*(σελ. 71- 80). Πάτρα
- Αναστασιάδης, Π., Μικρόπουλος Α., Σοφός, Α. & Φραγκάκη, Μ. (2010). *Ο διαδραστικός πίνακας στη σχολική τάξη: Παιδαγωγικές προσεγγίσεις - διδακτικές εφαρμογές*. Αθήνα: Παιδαγωγικό Ινστιτούτο.
- Βοσνιάδου, Σ. (2006). *Σχεδιάζοντας περιβάλλοντα μάθησης υποστηριζόμενα από τις Σύγχρονες Τεχνολογίες*. Αθήνα: GUTENBERG.
- Γκιμπερίτης, Β. (2003). *Η κοινωνία της πληροφορίας στην εκπαίδευση - Εφαρμογές Πληροφορικής*. Θεσσαλονίκη: Γκιούρδας Β. Εκδοτική
- Γρηγοριάδου, Μ., Γόγουλου, Α. & Γουλή, Ε. (2002). *Εναλλακτικές Διδακτικές Προσεγγίσεις σε Εισαγωγικά Μαθήματα Προγραμματισμού. Στο Α. Δημητρακοπούλου (Επιμ.), Πρακτικά 3^{ου} Πανελληνίου Συνεδρίου με Διεθνή Συμμετοχή «Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση», 239-248. Ρόδος.*
- Δημητρακοπούλου Α., (2004). *Οι Τεχνολογίες της πληροφορίας και των Επικοινωνιών στην πρωτοβάθμια Εκπαίδευση, Διδακταλική Ομοσπονδία Ελλάδας/Ινστιτούτο παιδαγωγικών Ερευνών και Μελετών*. Αθήνα.
- Δημητριάδης, Σ. (2015). *Θεωρίες Μάθησης και Εκπαιδευτικό Λογισμικό*. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/3397>
- Εφόπουλος, Β., Ευαγγελίδης, Γ., Δαγδιλέλης, Β. & Κλεφτοδήμος, Α. (2005). *Οι Δυσκολίες των Αρχάριων Προγραμματιστών. Στο Α. Τζιμογιάννης (Επιμ.), Πρακτικά 3^{ου} Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής»*. Κόρινθος: Πανεπιστήμιο Πελοποννήσου.
- Ζωγόπουλος, Ε. (2001). *Νέες Τεχνολογίες και Μέσα Επικοινωνίας στην εκπαιδευτική διαδικασία*. Αθήνα: Κλειδάριθμος.
- Ιωαννίδης, Γ. Σ. & Παναγιωτακόπουλος, Χ. Θ. (1994). *Ο ηλεκτρονικός υπολογιστής - Μία...κατάδυση ως τον πυθμένα, Computers: Inside- out*. Αθήνα: Εκδόσεις Καστανιώτη.
- Καλούρη-Αντωνοπούλου Ο. & Σιγάλας Χ. (2006). *Γενική Διδακτική Μεθοδολογία. Γενικά Ψυχοπαιδαγωγικά Θέματα*, Αθήνα: Μεταίχμιο.
- Καράμνης, Ι. (2006). *Διδασκαλία και Μάθηση με την Αξιοποίηση του Διαδικτύου: Ο ρόλος του εκπαιδευτικού*. Αθήνα: Ατραπός.

- Κατωπόδης, Α. (2010). *Πρόταση διδασκαλίας με τη χρήση των νέων τεχνολογιών της Δ.Ε. της Γ' Γυμνασίου, "Μέγας Βασίλειος: Ένας κορυφαίος Ιεράρχης", Κοινωνία ΝΓ' (2010), σ. 343-353.*
- Κόμης Β. & Μικρόπουλος Α. (2001). *Πληροφορική στην Εκπαίδευση*. Πάτρα: Ε.Α.Π.
- Κόμης, Β. (2004). *Εισαγωγή στις εκπαιδευτικές εφαρμογές των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών*. Αθήνα: Εκδόσεις Ν. Τεχνολογιών.
- Κόμης, Β. (2005). *Εισαγωγή στη Διδακτική της Πληροφορικής*. Αθήνα: Εκδόσεις Κλειδάριθμος.
- Κόμης Β. (2015). *Εφαρμογές των Τεχνολογιών της Πληροφορίας και των Επικοινωνιών στη διδασκαλία και τη μάθηση, Ενότητα 2: Οι Τεχνολογίες της Πληροφορίας & των Επικοινωνιών ως εργαλεία με γνωστικό δυναμικό. Έκδοση: 1.0*. Πάτρα. (Από τη δικτυακή διεύθυνση: <https://eclass.upatras.gr/courses/PN1441>).
- Κυνηγός, Χ. & Δημαράκη, Ε. (2002). *Νοητικά εργαλεία και πληροφορικά μέσα. Παιδαγωγική αξιοποίηση σύγχρονης τεχνολογίας για τη μετεξέλιξη της εκπαιδευτικής πρακτικής*. Αθήνα: Καστανιώτη.
- Λαφατζή, Ι. Γ. (2005). *Νέες τεχνολογίες στην εκπαίδευση*. Θεσσαλονίκη: Αφοί Κυριακίδη
- Μαλλιαράκης, Χ., Ξυνόγαλος, Σ. & Σατρατζέμη, Μ. (2012). *Εκπαιδευτικά παιχνίδια για την εκμάθηση του προγραμματισμού. Πρακτικά 8ου Πανελλήνιου Συνεδρίου με Διεθνή Συμμετοχή «Τεχνολογίες της Πληροφορίας & Επικοινωνίας στην Εκπαίδευση»*. Βόλος: Πανεπιστήμιο Θεσσαλίας
- Μακράκης, Β. & Κοντογιαννοπούλου-Πολυδωρίδη, Γ. (1995). *Υπολογιστές στην εκπαίδευση: μια κριτική επισκόπηση στο διεθνή χώρο και στην Ελλάδα*. Αθήνα: Εθνικό Ίδρυμα Ερευνών
- Μαραγκός Κ. & Γρηγοριάδου, Μ. (2005). *Η δυναμική των ηλεκτρονικών παιχνιδιών στη μαθησιακή διαδικασία. Μία πρόταση αντιμετώπισης των μαθησιακών δυσκολιών στον προγραμματισμό των πινάκων. Πρακτικά 3ου Πανελλήνιου Συνεδρίου των Εκπαιδευτικών για τις ΤΠΕ «Αξιοποίηση των Τεχνολογιών της Πληροφορίας και της Επικοινωνίας στη Διδακτική Πράξη»*. Σύρος.
- Μαυροματίη, Μ. (2010). Τα βιντεοπαιχνίδια ως εργαλεία μάθησης. Από: http://lexifilia.blogspot.com/2012/10/blog-post_5047.html
- Μητροπούλου, Β. (2008). *Το Εκπαιδευτικό Λογισμικό στη διδασκαλία του μαθήματος των Θρησκευτικών*. Θεσσαλονίκη: Βάνιας.
- Μικρόπουλος, Τ. Α. (2006). *Ο υπολογιστής ως γνωστικό εργαλείο*. Αθήνα: Ελληνικά Γράμματα.
- Μικρόπουλος, Α. Τ. (2009). *Εκπαιδευτικό λογισμικό. Θέματα σχεδίασης και αξιολόγησης λογισμικού υπερμέσων*. Αθήνα: Κλειδάριθμος.
- Μπράτιτσης, Θ. (2013). *Η Πληροφορική στο Ελληνικό Σχολείο: Τάσεις, προσεγγίσεις, προοπτικές. Θέματα Επιστημών και Τεχνολογίας στην Εκπαίδευση, 6(3), 111-115*. Ανακτήθηκε από <http://earthlab.uoi.gr/thete/index.php/thete>

- Νιάρου, Β. & Γρουσουζιάκου, Ε. (2007). *Η αξιοποίηση των τεχνολογιών της πληροφορίας και των επικοινωνιών στη διδακτική πράξη. Ο διαδραστικός πίνακας στην εκπαίδευση. 4^ο Πανελλήνιο Συνέδριο Εκπαιδευτικών για τις ΤΠΕ*. Σύρος.
- Νικηφορίδου, Ζ., & Παγγέ, Τ. (2011). *Ψηφιακό παιχνίδι στην Προσχολική Ηλικία. 6^ο Διεθνές Συνέδριο για την Ανοιχτή και Εξ αποστάσεως εκπαίδευση*. Λουτράκι.
- Ξυνόγαλος, Σ. & Σατρατζέμη, Μ. (2004). *Η Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό: Προβλήματα και Μεθοδολογίες για την Αντιμετώπισή τους. Πρακτικά 4ου Πανελλήνιου Συνεδρίου με Διεθνή Συμμετοχή «Οι Τεχνολογίες της Πληροφορίας και της Επικοινωνίας στην Εκπαίδευση», 133-142 (τόμος Β)*. Αθήνα.
- Παναγιωτακόπουλος, Χ., Πιερρακέας, Χ. & Πιντέλας, Π. (2003). *Το εκπαιδευτικό λογισμικό και η αξιολόγησή του*. Αθήνα: Μεταίχμιο.
- Πανουτσόπουλος, Η. (2010). *Αξιοποίηση των Ψηφιακών Παιχνιδιών στο πλαίσιο του Αναλυτικού Προγράμματος Σπουδών των Σχολείων Δευτεροβάθμιας Εκπαίδευσης. Μεταπτυχιακή Διπλωματική Εργασία. Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς*.
- Παπαϊωάννου, Γ. & Μόχλα Α. (2006). *Μάθετε εύκολα το Microsoft Excel*. Λάρισα: Paradox Interactive. (<http://www.paradoxinteractive.gr/προϊόν/μάθετε-εύκολα-το-microsoft-excel/>)
- Πατρικάκης Χ., Τουμανίδης Λ. & Κόγιας Δ. *Οδηγίες χρήσης και σύντομη εισαγωγή στην πλατφόρμα Greenfoot*. Τμήμα Ηλεκτρονικών Μηχανικών: ΤΕΙ Πειραιά.
- Ράπτης, Α. & Ράπτη Α. (1998). *Πληροφορική και Εκπαίδευση – Συνολική προσέγγιση*. Αθήνα: Ιδιωτική
- Ράπτης, Α. & Ράπτη Α. (2006). *Μάθηση και Διδασκαλία στην Εποχή της Πληροφορίας, Παιδαγωγικές Δραστηριότητες, Τόμος Β΄*. Αθήνα: Έκδοση συγγραφέων.
- Σιακαβάρας Ι., *Εισαγωγή στον αντικειμενοστρεφή προγραμματισμό με τη βοήθεια παιχνιδιών: Η περίπτωση του Greenfoot*. Υπουργείο Παιδείας, Ινστιτούτο Εκπαιδευτικής Πολιτικής, <http://aesop.iep.edu.gr/node/15856>
- Σολομωνίδου, Χ. & Σταυρίδου, Ελ. (1994). *Σύγχρονη εκπαιδευτική τεχνολογία: Δυνατότητες και προοπτικές για την επίλυση προβλημάτων της εκπαίδευσης, Παιδαγωγική Επιθεώρηση, τ.20-21, σελ. 69-91*.
- Σολομωνίδου, Χ. (1999). *Εκπαιδευτική Τεχνολογία. Μέσα, υλικά, διδακτική χρήση και αξιοποίηση*. Αθήνα: Εκδόσεις Καστανιώτη.
- Σολομωνίδου, Χ. (2006). *Νέες τάσεις στην εκπαιδευτική τεχνολογία. Εποικοδομισμός και σύγχρονα περιβάλλοντα μάθησης*. Αθήνα: Μεταίχμιο.
- Σολομωνίδου, Χ. (2009). *Η χρήση του υπολογιστή στο σύγχρονο σχολείο*. Πανεπιστημιακές Εκδόσεις Θεσσαλίας.

- Τάσσης, Ο. (2014). *Οι σχέσεις των εκπαιδευτικών με τις Τεχνολογίες της Πληροφορίας και Επικοινωνιών στο σχολείο. Έρκυνα, Επιθεώρηση Εκπαιδευτικών-Επιστημονικών Θεμάτων. Τεύχ. 1., σσ.200-215*
- Τατόγλου, Χ. (2018). *Αξιολόγηση του Greenfoot ως εργαλείου για την εισαγωγή στον προγραμματισμό παιχνιδιών. Θεσσαλονίκη: Πανεπιστήμιο Μακεδονίας.*
- Τζιμογιάννης Α. & Γεωργίου Β. (1998). *Η αναγκαιότητα της διδασκαλίας του προγραμματισμού Η/Υ στη δευτεροβάθμια εκπαίδευση ως μεθοδολογία επίλυσης προβλημάτων. Το παράδειγμα των πινάκων. Πρακτικά Διημερίδας Πληροφορικής «Η Πληροφορική στη Δευτεροβάθμια Εκπαίδευση». Αθήνα: ΕΠΥ.*
- Τζιμογιάννης Α. & Κόμης Β. (1999). *Επίλυση προβλημάτων σε προγραμματιστικό περιβάλλον: η οικοδόμηση της δομής ελέγχου από τους μαθητές του Ενιαίου Λυκείου, στο Α. Κόλλιας, Α. Μαργετουσάκη & Π. Μιχαηλίδης (επιμ.), Πρακτικά 4^{ου} Πανελλήνιου Συνεδρίου «Διδακτική των Μαθηματικών και Πληροφορική στην Εκπαίδευση», 243-249. Ρέθυμνο.*
- Χριστοπούλου, Α. (2000). *Γιατί αποτυγχάνουν οι μαθητές στην επίλυση προβλημάτων σε προγραμματιστικό περιβάλλον. Πρακτικά 2^{ου} Συνέδριου ΕΤΠΕ. Πάτρα.*

ΞΕΝΟΓΛΩΣΣΗ

- Alessi, S.M. & Trollip, S.R. (2000). *Multimedia for learning: Methods and development*. Boston: MA.
- Anderson, J. & van Weert, T. (Eds). 2002. *Information and Communication Technology in Education: A Curriculum for Schools and Programme of Teacher Development*. UNESCO, Paris
- Bayman, P. & Mayer, R.E. (1988). *Using conceptual models to teach BASIC computer programming. Journal of Educational Psychology, 80(3), 291–298.*
- Beck, K. (1993). *CRC: Finding objects the easy way. Object Magazine, 3(4), 42–44.*
- Cheng, M., Chen, J., Chu, S. et al. (2015). *The use of serious games in science education: a review of selected empirical research from 2002 to 2013. J. Comput. Educ. 2, 353–375.*
- Cohen, L., Manion, L. & Morrison, K.R.B. (2011). *Research methods in education*. Oxon, UK: Routledge
- Connolly, Thomas & Stansfield, M. & Hainey, Thomas. (2008). *Development of a general framework for evaluating games-based learning. Proceedings of the European Conference on Games-based Learning. 105-114.*
- Crook, C., Harrison, C., Farrington-Flint, L., Tomás, C. & Underwood, J. (2010). *The Impact of Technology: Value-added classroom practice*. Coventry: Becta.

- Cudworth, A. L. (1996). *Simulation and Games. International Journal of Educational Technology, Second Edition*. T. Plomp and D. P. Ely (Eds), Oxford: Pergamon.
- Dempsey, J. V., Haynes, L. L., Lucassen, B.A., Casey, M. S. (2002). *Forty simple computer games and what they could mean to educators. Simulation & Gaming. 33(2), 157-168.*
- Du Boulay, J. B. H. (1989). *Some difficulties of learning to program. In E. Soloway and J.C. Spohrer, (Eds.), Studying the Novice Programmer (pp.283-299 & pp. 431-436).* Hillsdale NJ: Lawrence Erlbaum Associates.
- Fincher, S. (2010). *Comparing Alice, Greenfoot & Scratch*. SIGCSE
- Gee, J. P. (2003). *What Video Games have to teach us about learning and literacy?* New York: Palgrave Macmillan.
- Grüsser, S., Thalemann, R. & Griffiths, M. (2007). *Excessive Computer Game Playing: Evidence for Addiction and Aggression? Cyberpsychology & behavior: the impact of the Internet, multimedia and virtual reality on behavior and society.* (10.1089/cpb.2006.9956)
- Gunter, B. (2005). *Psychological effects of video games. In Raessens, J. F. F., Goldstein, J. (eds.), Handbook of Computer Game Studies, pp. 145-160.* Cambridge, MA: MIT Press
- Gunter, G.A. and Kenny, R. (2005). Thinking Out of the Hexagon - Digital Media in the Classroom. Association for Educational Communications and Technology International Annual (AECT). Orlando, Florida.
- Kirriemuir, J. K. & McFarlane, A. (2003). *Use of Computer and Video Games in the Classroom. Proceedings of the Level Up Digital Games Research Conference.* Netherlands: Universiteit Utrecht,
- Kölling M. (2010). *Introduction to Programming with Greenfoot*. Prentice Hall
- Loftus, G. R. & Loftus, E. F. (1983). *Mind at play: the psychology of video games.* New York, NY: Basic Books.
- Major, L., Kyriacou, Th. & Brereton, P. (2012). *Teaching Novices Programming Using a Robot Simulator: Case Study Protocol.* (10.13140/2.1.4831.5844).
- McGill, T. J. & Violet, S. E. (1997). *A conceptual framework for analyzing students' knowledge of programming. Journal of Research on Computing in Education, 29 (3), 276-297.* (10.1080/08886504.1997.10782199)
- Papert, S. (1993). *The children's machine. Rethinking School in the Age of the Computer.* New York: BasicBooks, A Division of HarperCollins Publishers, Inc.
- Pivec, M., Dziabenko, O., Schinnerl, I. (2004). *Aspects of Game-based Learning. I-Know. J. UCS Proceedings of I-Know, pp. 217-224.*

- Pivec, M., Koubek, A. & Dondi, C. (2004). *Guidelines for Game-Based Learning*. Lengerich, Germany: Pabst Science Publishers
- Prensky, M. (2003). *Digital game-based learning*. McGraw-Hill.
- Prensky, M. (Μετ. Παπασταύρου Ν. & Παπασταύρου Κ.) (2009). *Μάθηση βασισμένη στο ψηφιακό παιχνίδι. Αρχές, δυνατότητες και παραδείγματα εφαρμογής στην εκπαίδευση και την κατάρτιση*. Αθήνα: Μεταίχμιο.
- Ramadhan, Haider. (2000). *Programming by discovery. Journal of Computer Assisted Learning. Vol 16 (1). 83-93.*
- Rosas, R., Nussbaum, M., Cumsille, P., Marianov, V., Correa, M., et al. (2003). *Beyond Nintendo: design and assessment of educational video games for first and second grade students. Computers & Education, 40, 71-94.*
- Soloway, E. & J. C. Spohrer J. C. (2013). *Studying the Novice Programmer. Interacting with Computers Series*. Psychology Press.
- Utting, I., Cooper, St., Kölling, M., Maloney, J. & Resnick, M. (2010). *Alice, Greenfoot, and Scratch - A Discussion. ACM Transactions on Computing Education (TOCE).*
- Vos, N., van der Meijden, H. & Denessen, E. (2011). *Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. Computers & Education, 56, 127-137.*
- Williams, L. & Kessler, R. (2000). *All I really need to know about pair programming I learned in kindergarten. Communications of the ACM. (43. 108-114. 10.1145/332833.332848).*
- Woei, L. S., Othman, I. H. & Man C. K. (2013). *Learning Programming Using Objects-First Approach Through Folktales, 2nd International Conference on Interactive Digital Media, Jurnal Teknologi 75 (3).*
- Xinogalos, S., Satratzemi, M., & Malliarakis, C. (2015). *Microworlds, games, animations, mobileapps, puzzle editors and more: What is important for an introductory programmingenvironment? Education and Information Technologies, 1-32.*

ΔΙΚΤΥΑΚΟΙ ΤΟΠΟΙ

<https://en.wikipedia.org/wiki/Greenfoot>
https://el.wikipedia.org/wiki/Τεχνολογία_πληροφοριών
<http://www.greenfoot.org>
<http://www.greenfoot.org/files/javadoc/>

<https://greenroom.greenfoot.org>

<http://www.gsae.edu.gr/el/>

<http://www.oracle.com/webfolder/technetwork/tutorials/OracleAcademy/GreenfootSelfStudyV1/obe.html>

<http://www.pi-schools.gr/programs/depps/>

http://users.sch.gr/tsakarak/Yliko_Blog/Panepistimio/PAIDAG/Dialeksi_1C.pdf

ΠΑΡΑΡΤΗΜΑ Ι – ΕΚΠΑΙΔΕΥΤΙΚΟ ΣΕΝΑΡΙΟ

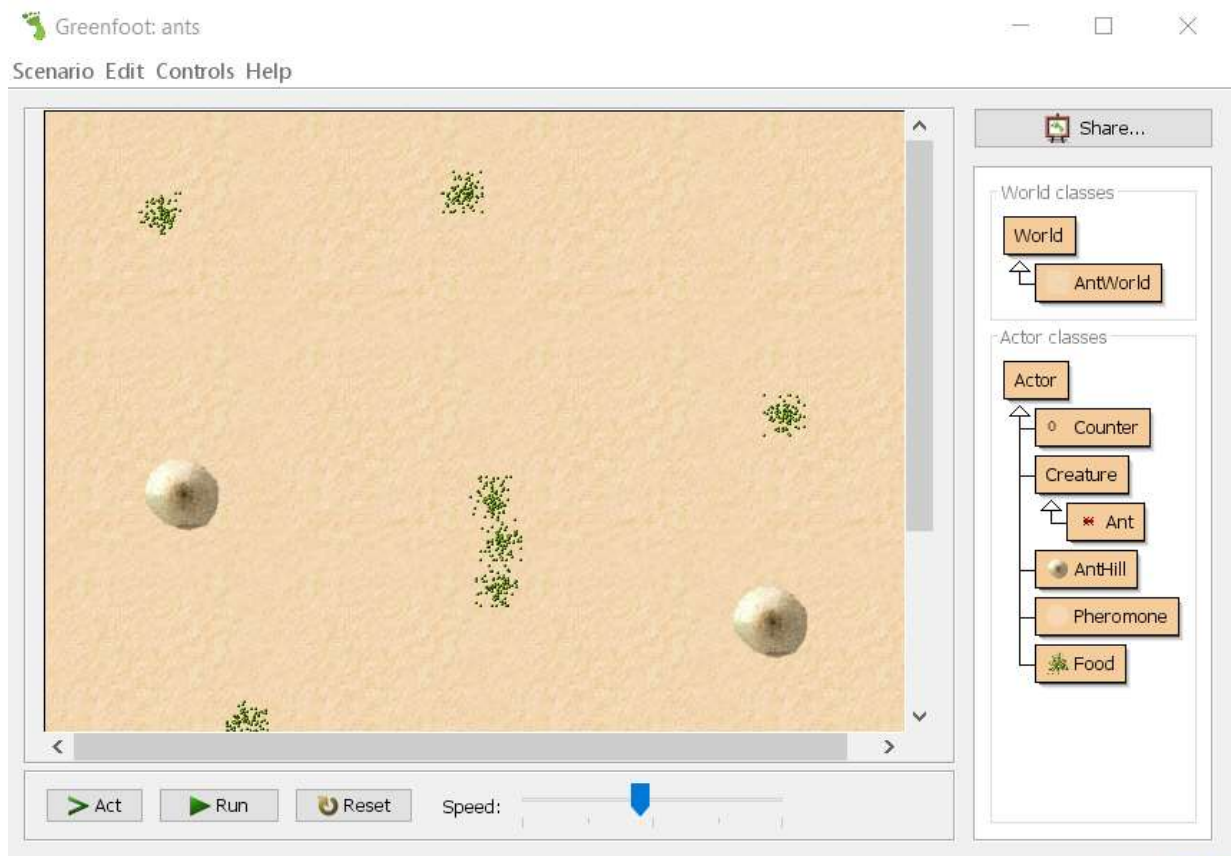
Εγκατάσταση εφαρμογής



Κατεβάστε και εγκαταστήστε την εφαρμογή GreenFoot από τη διεύθυνση

<http://www.greenfoot.org/download/files/Greenfoot-windows-310.msi>

Ακολουθήστε τις οδηγίες και στην ερώτηση για πρόσβαση στο διαδίκτυο απαντήστε καταφατικά.

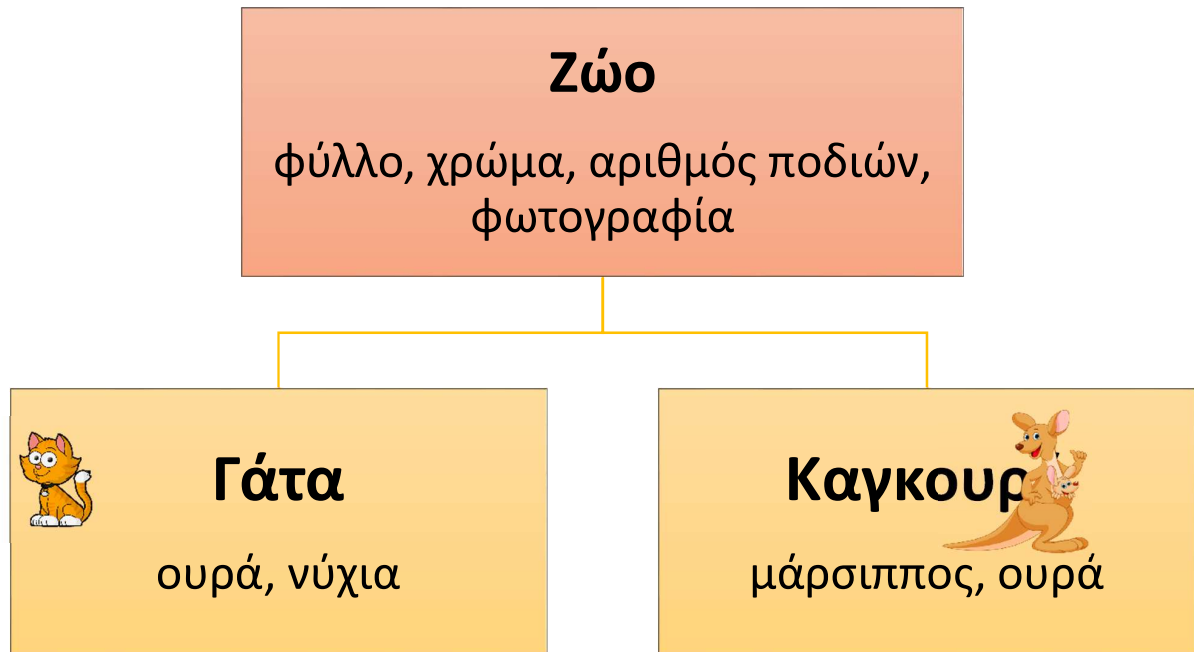


Αν επιθυμείτε να αλλάξετε τη γλώσσα διεπαφής, επιλέξτε Edit→Preferences, επιλέξτε την καρτέλα «Interface» και «Greek». Για να ενεργοποιηθούν οι αλλαγές, κάντε επανεκκίνηση της εφαρμογής.

Θεωρία

Στον αντικειμενοστραφή προγραμματισμό η περιγραφή του προβλήματος γίνεται με βάση τα αντικείμενα (objects), τα οποία θεωρούνται οντότητες, που αναπαριστούν στοιχεία του πραγματικού χώρου. Με τον τρόπο αυτό, η επίλυση του προβλήματος πηγάζει άμεσα από το ίδιο το πρόβλημα.

Έτσι, η περιγραφή του προβλήματος μέσω του προγράμματος γίνεται με τους όρους του προβλήματος και όχι της μηχανής. Οι οντότητες έχουν συγκεκριμένες ιδιότητες όπως χρώμα, διαστάσεις, βάρος, ηλικία, όνομα κτλ. Για παράδειγμα, το αντικείμενο «ζώο» είναι γενικό και αφορά όλα τα ζώα του πλανήτη. Όλα διαθέτουν μάτια, χρώμα, πόδια.



Παρόλα αυτά, ορισμένα ζώα έχουν περισσότερες ιδιότητες από αυτές του βασικού αντικειμένου. Για παράδειγμα, η γάτα διαθέτει επιπλέον «ουρά» και «νύχια», ενώ το καγκουρό «ουρά» και «μάρσιππο». Η γάτα και το καγκουρό είναι δύο νέα αντικείμενα, τα οποία, όμως, κληρονομούν τις ιδιότητες του αρχικού αντικειμένου στην ιεραρχία και πάνω σε αυτό προσθέτουν και κάποιες άλλες. Σε γενικές γραμμές, η παραπάνω περιγραφή αποδίδει την έννοια των αντικειμένων. Στην ορολογία του αντικειμενοστρεφούς προγραμματισμού, κάθε αντικείμενο ονομάζεται κλάση (class), μέσα στην οποία ενσωματώνονται τόσο οι ιδιότητες (attributes), όσο και οι συμπεριφορές, με τις οποίες μπορούμε να αξιοποιήσουμε αυτές τις ιδιότητες, για παράδειγμα να εμφανίσουμε όλες τις φωτογραφίες των ζώων ή να βρούμε ποιο ζώο έχει μάρσιππο. Αυτές οι συμπεριφορές υλοποιούνται στην ορολογία του αντικειμενοστρεφούς προγραμματισμού με τις κλασικές συναρτήσεις, όπως και στον διαδικαστικό/συναρτησιακό προγραμματισμό, μόνο που εδώ ονομάζονται μέθοδοι (methods). Δείτε τη δήλωση της κλάσης «ζώα», κωδικοποιημένη στη γλώσσα προγραμματισμού Java.

```
1. public class animal{  
2.  
3.     int sex;  
4.     int color;
```

```

5.     int legs_number;
6.     ImageIcon photo;
7.
8.
9. };

```

Οι ιδιότητες `sex`, `color`, `legs_number` και `photo` πρέπει να ορίσουν το είδος των δεδομένων, που πρόκειται να αποθηκεύσουν στη μνήμη του υπολογιστή. Η δήλωση `int` σημαίνει ακέραιος αριθμός και είναι σύντμηση της λέξης *Integer*. Η *ImageIcon* είναι μια πιο περίπλοκη δομή, που μπορεί να αποθηκεύσει εικόνες. Η δήλωση *public* ενημερώνει ότι όλοι έχουν πρόσβαση στη νέα κλάση (σημαίνει ότι υπάρχει προστασία ως προς τη δυνατότητα πρόσβασης, αλλά δεν θα ασχοληθούμε με αυτό). Αντίστοιχα, έχουμε κλάσεις για τη γάτα και το καγκουρό.

```

1. public class cat{
2.
3.     boolean tail;
4.     int nails;
5.
6. };
7.
8. public class kangaroo{
9.
10.    boolean marsupium;
11.    boolean tail;
12.
13. };

```

Η δήλωση *boolean* σημαίνει ότι μπορεί να αποθηκεύσει μια τιμή αληθή (`true`) ή ψευδή (`false`). Για παράδειγμα, στην ερώτηση «η γάτα έχει νύχια» η απάντηση θα ήταν «`true`», δηλαδή αληθής. Η δυνατότητα να κληρονομήσουν οι κλάσεις γάτα και καγκουρό τις αρχικές ιδιότητες ονομάζεται κληρονομικότητα (Inheritance). Το μόνο, που χρειάζεται να αλλάξουμε από τον κώδικα παραπάνω, είναι τη δήλωση «*extends Animal*».

```

1. public class cat extends animal{
2.
3.     boolean tail;
4.     int nails;
5.
6. };
7.
8. public class kangaroo extends animal{
9.
10.    boolean marsupium;
11.    boolean tail;
12.
13. };

```

Όπως ίσως διαπιστώσατε οι ιδιότητες δεν είναι τίποτε παραπάνω από δηλώσεις μεταβλητών μέσα σε μία κλάση. Όλες οι μεταβλητές έχουν ένα όνομα και έναν τύπο δεδομένων, που τις αντιπροσωπεύει. Οι τύποι δεδομένων απλά υποδηλώνουν τι είδους δεδομένα πρόκειται να

καταχωρηθούν στη μνήμη του υπολογιστή. Πρόκειται για παράδειγμα να αποθηκεύσουμε αριθμούς ή κείμενα; Αν είναι αριθμοί, θα είναι ακέραιοι ή και δεκαδικοί; Αν είναι ακέραιοι, πόσο μεγάλοι θα είναι; Δηλαδή, με λίγα λόγια, ποια κλίμακα αριθμών μπορούν να αποθηκεύσουν;

Θυμηθείτε ότι η μνήμη έχει περιορισμένο μέγεθος και όσο πιο συνετά τη χρησιμοποιήσετε, τόσο καλύτερη απόδοση θα έχετε στην ταχύτητα της εκτέλεσης του προγράμματος. Αν, για παράδειγμα, χρειάζεστε μια ιδιότητα, για να σας ενημερώνει πόσα νύχια έχει ένα ζώο, δεν χρειάζεται να δηλώσετε ως τύπο δεδομένων έναν ακέραιο αριθμό (ο δεκαδικός δεν μας κάνει, αφού ένα ζώο δεν μπορεί να έχει για παράδειγμα 4,5 νύχια), ο οποίος θα έχει αποθηκευτική ικανότητα μιας κλίμακας από μηδέν μέχρι κάτι δισεκατομμύρια. Είναι σπατάλη χώρου, αλλά και χρόνου κατά την εκτέλεση του προγράμματος. Έτσι, μία κλίμακα από το μηδέν μέχρι το 255 είναι υπεραρκετή - δυστυχώς δεν μπορούμε να έχουμε μικρότερη κλίμακα από αυτή χωρίς δική μας παρέμβαση.

Το σημαντικότερο σε όλα αυτά είναι ότι μπορείτε να δημιουργήσετε όσες γάτες και καγκουρό θέλετε. Αυτά ονομάζονται *στιγμιότυπα* των αντίστοιχων κλάσεων *cat* και *kangaroo* και κάθε ένα έχει τη δική του φωτογραφία, χρώμα, φύλο κτλ.

Πέρα, όμως, από τις ιδιότητες (attributes), που έχει η κάθε κλάση, δηλαδή αν κάποιο ζώο έχει ουρά ή νύχια κτλ, θα πρέπει να έχουμε τη δυνατότητα να τις αξιοποιήσουμε με κάποιο τρόπο. Θα επιθυμούσαμε, για παράδειγμα, να δούμε τη φωτογραφία του ζώου ή να μάθουμε ποια είναι τα ιδιαίτερα χαρακτηριστικά του όσον αφορά στο χρώμα, στο φύλλο κτλ (εμφάνιση δηλαδή όλων των ιδιοτήτων που έχει). Αυτό επιτυγχάνεται με τις συμπεριφορές της κλάσης, οι οποίες υλοποιούνται στην πράξη με απλές συναρτήσεις, όπως στον κλασικό προγραμματισμό, μόνο που εδώ ονομάζονται μέθοδοι.

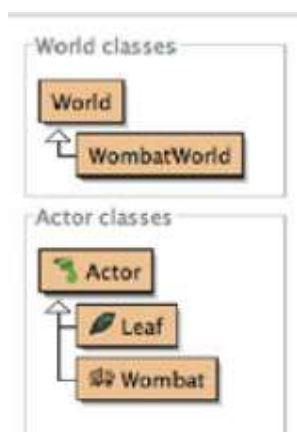
```
1. void setColor(int setcolor)
2. {
3.     color = setcolor;
4. }
5.
6. int getColor()
7. {
8.     return color;
9. }
10.
11. void printColorToScreen()
12. {
13.     System.out.println("Color code = " + color);
14. }
```

Παραπάνω φαίνονται τρεις μέθοδοι με τα ονόματα *setColor*, *getColor* και *printColorToScreen*. Στην πρώτη μέθοδο, με όνομα *setColor*, ορίζεται η ιδιότητα *color* της κλάσης *animal*. Ο ακέραιος αριθμός, που δίνεται μέσω της παραμέτρου *setColor*, αποθηκεύεται στο στιγμιότυπο της κλάσης, που επιλέχθηκε και μπορεί να είναι μια από τις *animal*, *cat* και *kangaroo*. Αν θέλετε να ανακτήσετε το

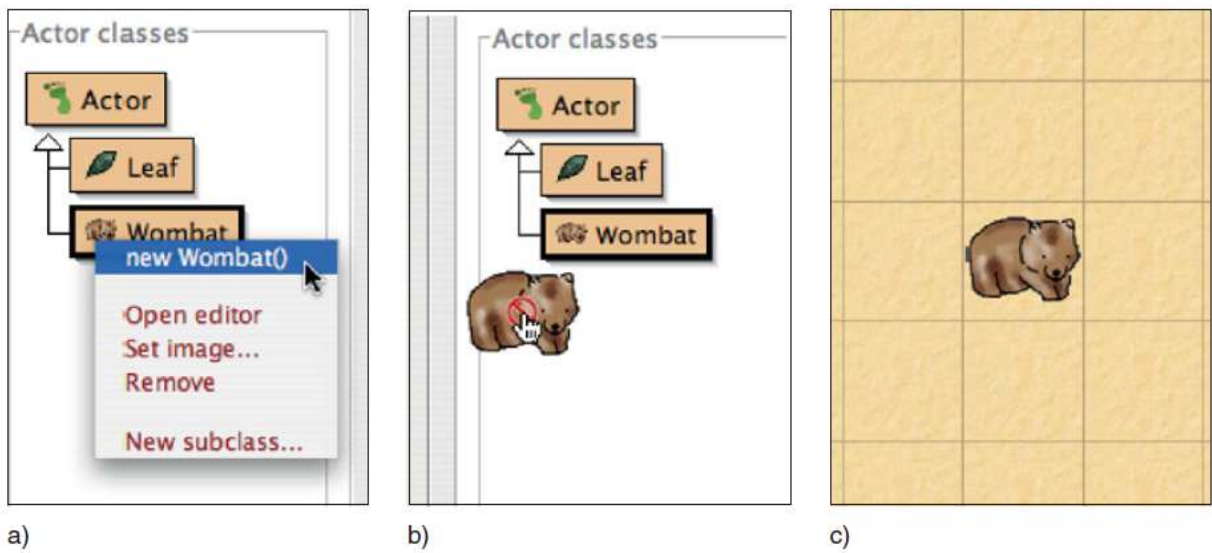
χρώμα, που μόλις αποθηκεύσατε, τότε μπορείτε να χρησιμοποιήσετε τη δεύτερη μέθοδο με το όνομα *getColor*. Τέλος, αν θέλετε να τυπώσετε στην οθόνη το χρώμα, που έχει το επιλεγμένο ζώο, τότε απλά καλέστε τη μέθοδο *printColorToScreen*.

Όπως ίσως παρατηρήσατε, οι μέθοδοι άλλοτε χρειάζονται κάποια είσοδο, άλλοτε επιστρέφουν κάτι στην έξοδο και ορισμένες φορές και τα δύο ταυτόχρονα. Η είσοδος εμφανίζεται ακριβώς δεξιά από το όνομα με το άνοιγμα των παρενθέσεων. Κάθε μια από αυτές ονομάζεται παράμετρος και διαχωρίζονται με κόμμα. Παράδειγμα *setFeatures(int color, int legs, Boolean hair)*. Για να κάνετε χρήση της τυχαίας μεθόδου *setFeatures*, απλά πληκτρολογείτε το όνομα και μέσα στις παρενθέσεις πληκτρολογείτε τις πραγματικές τιμές, διαχωρίζοντάς τις, όπως προαναφέρθηκε, με κόμμα. Έστω ότι θέλετε το χρώμα με τον κωδικό μηδέν, που είναι το μαύρο, τέσσερα πόδια συνολικά και ναι στην ιδιότητα των μαλλιών. Πληκτρολογήστε απλά *setFeature(0,4,true)*.

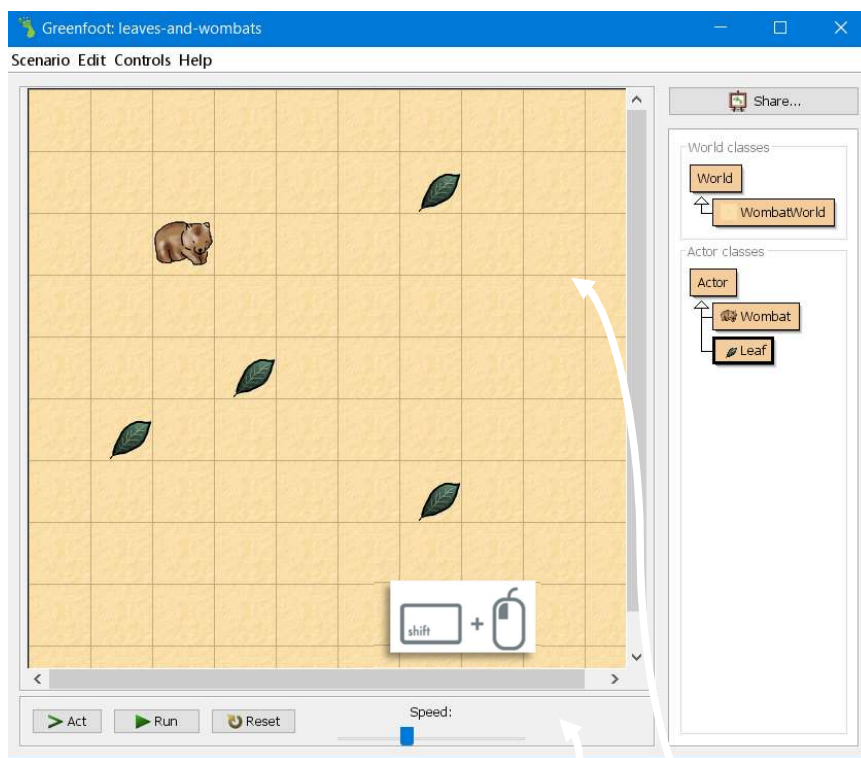
Σενάριο



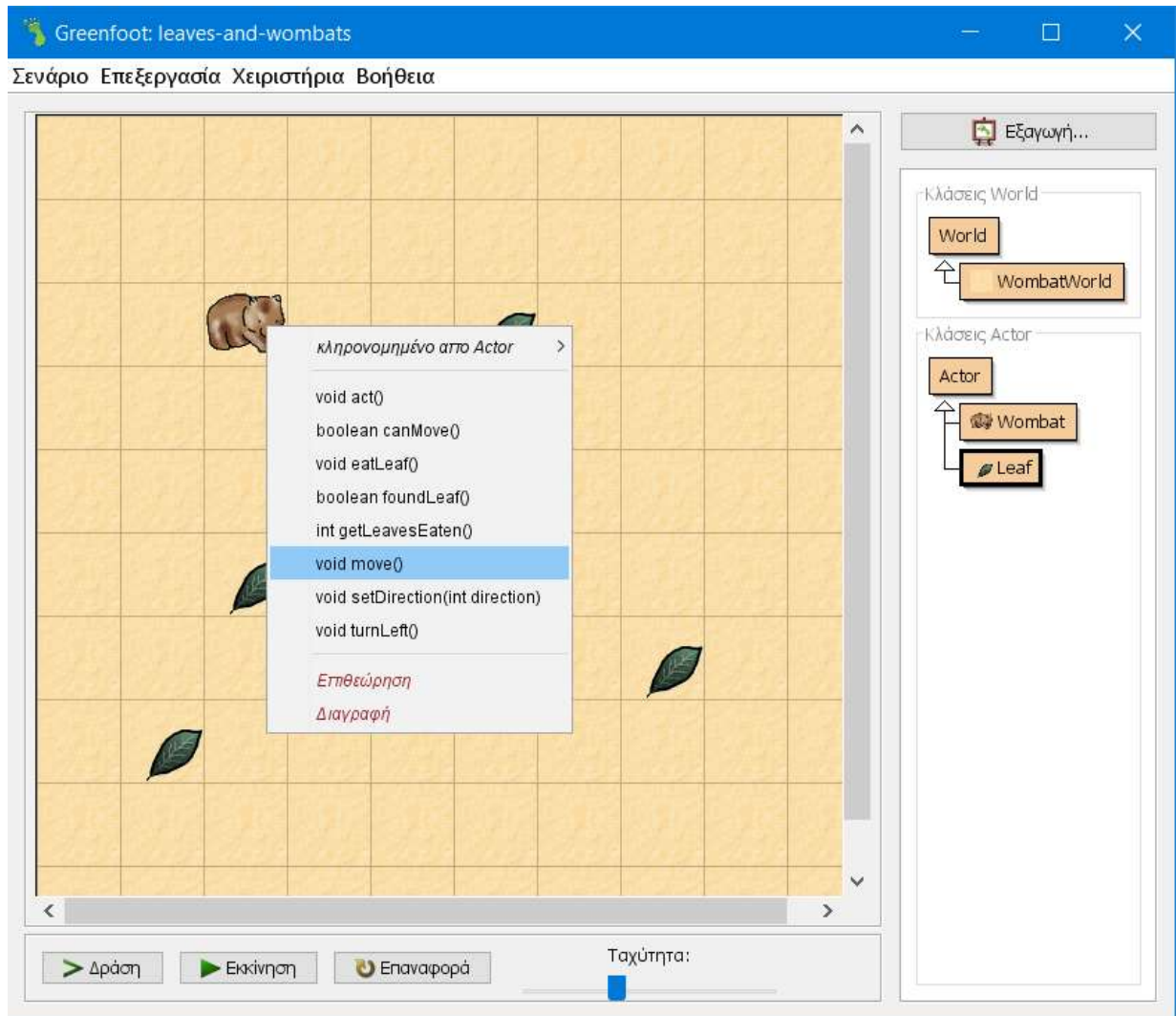
Ανοίξτε το επισυναπτόμενο αρχείο με το όνομα *Wombat*. Δείτε στο δεξιό μέρος της εφαρμογής *Greenfoot* τις δυο αρχικές κλάσεις *World* και *Actor*. Οι υποκλάσεις *WombatWorld* και *Leaf* και *Wombat* κληρονομούν όλες τις ιδιότητες των κλάσεων γονέων *World* και *Actor* αντίστοιχα. Τα βελάκια δείχνουν ακριβώς αυτό το χαρακτηριστικό της κληρονομικότητας. Αν κάνετε δεξί κλικ επάνω στο *Wombat*, θα εμφανιστεί το μενού της επιλεγμένης κλάσης (κάτω φωτογραφία σχήμα a). Η πρώτη επιλογή με τον τίτλο «*new Wombat()*» δημιουργεί ένα νέο στιγμιότυπο αυτής της κλάσης μέσα στον κόσμο (η μεγάλη οθόνη αριστερά των κλάσεων). Θα δείτε το εικονίδιο του γλυκού αυτού μαρσιποφόρου, που ζει στην Αυστραλία να ακολουθεί τον δείκτη του ποντικιού σας (σχήμα b). Κάντε κλικ κάπου μέσα στον κόσμο σας (σχήμα c).



Εφόσον δημιουργήσατε μια κλάση, μπορείτε να δημιουργήσετε όσα στιγμιότυπα επιθυμείτε. Επαναλάβετε την ίδια διαδικασία με τα φύλλα (τοποθετήστε τουλάχιστον τέσσερα). Για μεγαλύτερη ευκολία, μπορείτε απλά να επιλέξετε μια κλάση με απλό κλικ. Στη συνέχεια, κρατήστε πατημένο το πλήκτρο [Shift] και κάντε κλικ επάνω στον κόσμο.



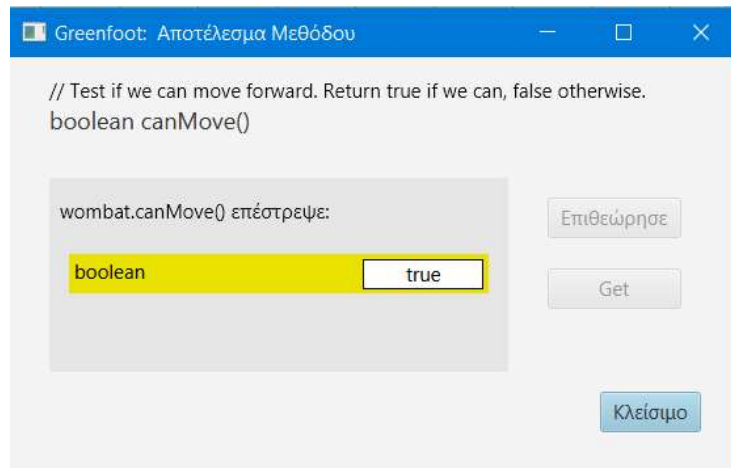
Τώρα που τοποθετήθηκαν όλα τα στιγμιότυπα των αντικειμένων μέσα στην κεντρική σκηνή, η οποία ονομάζεται «κόσμος», μπορείτε να αλληλοεπιδράσετε με αυτά, κάνοντας δεξί κλικ επάνω τους.



Το μενού, που θα ανοίξει, θα εμφανίσει όλες τις συμπεριφορές – λειτουργίες του συγκεκριμένου στιγμιότυπου (θυμηθείτε ότι κάθε στιγμιότυπο μπορεί να έχει τη δική του ξεχωριστή συμπεριφορά, παρότι προέρχονται όλα από την ίδια αρχική κλάση). Η επιλογή «κληρονομημένο από Actor» εμφανίζει όλες τις συμπεριφορές και ιδιότητες, που κληρονόμησε η κλάση Wombat από την αρχική κλάση Actor (το ζώο στο παράδειγμα της θεωρίας παραπάνω), ενώ όλες οι υπόλοιπες είναι οι καινούργιες συμπεριφορές, που αντιπροσωπεύουν μόνο την κλάση Wombat. Οι δύο τελευταίες επιλογές «Επιθεώρηση» και «Διαγραφή» δεν είναι συμπεριφορές και θα αναλυθούν παρακάτω.

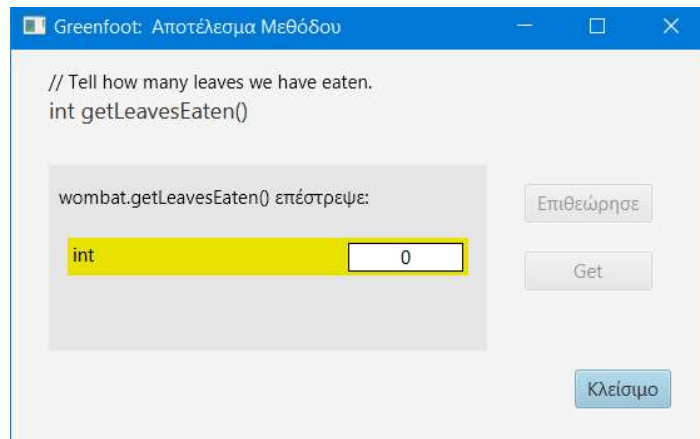
Κάντε κλικ επάνω στη συμπεριφορά «void move()» και παρατηρήστε. Δοκιμάστε το μερικές φορές και μετά δοκιμάστε τη συμπεριφορά «void turnLeft()». Αυτό που είδατε, ήταν η συμπεριφορά της κίνησης και της στροφής δεξιά μέσα στον κόσμο. Το void μπροστά, όπως αναφέρθηκε και στη

θεωρία, απλά ενημερώνει ότι δεν επιστρέφει καμία τιμή ως έξοδο μετά την εκτέλεση της συγκεκριμένης μεθόδου – συμπεριφοράς και οι κενές παρενθέσεις ότι δεν δέχονται τίποτα ως είσοδο.



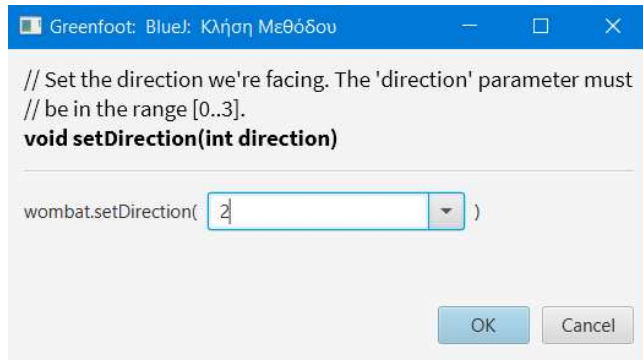
Στη συνέχεια, κάντε κλικ στη μέθοδο «boolean canMove()». Θα εμφανίσει το παράθυρο, που βλέπετε δίπλα σας. Εδώ η μέθοδος canMove είναι μία ερώτηση, η οποία ρωτάει την κλάση Wombat «Μπορεί να κινηθεί το μαρσιποφόρο;». Αυτή επιστρέφει μια τιμή, που μπορεί να είναι αληθής (true) ή ψευδής (false), εάν το όμορφο

μαρσιποφόρο μπορεί όντως να κινηθεί επάνω στον κόσμο. Η απάντηση είναι true στις περισσότερες περιπτώσεις, πράγμα που σημαίνει ότι ο ηθοποιός –μαρσιποφόρο, που τοποθετήσατε επάνω στον κόσμο, έχει τη δυνατότητα να κινηθεί, όπως άλλωστε και εσείς οι ίδιοι διαπιστώσατε προηγουμένως. Ωστόσο, όταν το μαρσιποφόρο φτάσει στο όρια του κόσμου σας, δεν μπορεί να προχωρήσει παραπέρα και σε αυτήν την περίπτωση η μέθοδος θα επιστρέψει false.



Δοκιμάστε τώρα να δείτε πόσα φύλλα έφαγε το μαρσιποφόρο σας μέχρι στιγμής. Κάντε δεξί κλικ επάνω σε αυτό και επιλέξτε «int getLeavesEaten()». Θα δείτε ότι πάντα επιστρέφει την τιμή μηδέν. Μπορείτε με τη χρήση των μεθόδων, που υπάρχουν να υποχρεώσετε το μικρό Wombat να φάει κάποιο φύλλο; Απλά μετακινήστε το Wombat επάνω σε

κάποιο φύλλο και κάντε κλικ επάνω στη μέθοδο «void eatLeaf()». Μόλις το κάνετε αυτό, το φυλλαράκι θα εξαφανιστεί από το συγκεκριμένο τετράγωνο και η μέθοδος «int getLeavesEaten()» θα σας εμφανίσει τον αριθμό 1.



Μια πιο εύκολη μέθοδος, για να περιστρέψετε το μικρό Wombat είναι η «void setDirection(int direction)». Εδώ, όπως βλέπετε, πρέπει ως είσοδο να δώσετε έναν αριθμό από το 0 έως το 3. Κάθε μία τιμή αντιπροσωπεύει και μια κατεύθυνση του μικρού ήρωα.

Όλα τα αντικείμενα μέσα στον κόσμο του GreenFoot έχουν μια μέθοδο ACT (δράση), που σημαίνει «δράσε», δηλαδή κάνε κάτι. Βέβαια είναι στο χέρι μας αν θέλουμε να την αξιοποιήσουμε. Επειδή όμως σε ένα παιχνίδι προφανώς και επιθυμούμε να έχουμε δράση, κίνηση και αποτελέσματα, χρησιμοποιείται σχεδόν πάντα. Αν κάνετε κλικ μια φορά στο κουμπί αυτό, τότε ο μικρός ήρωας θα ακολουθήσει τις εξής εντολές :

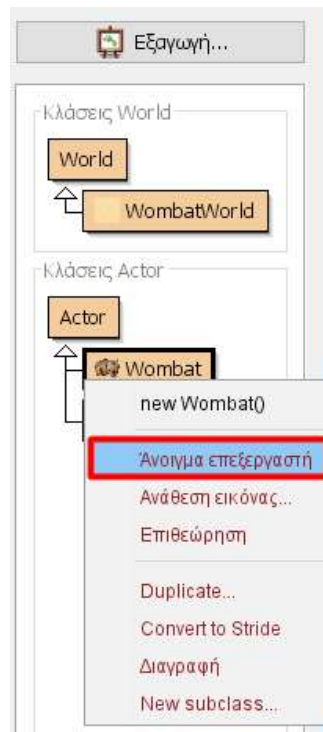
- Αν βρίσκεται μέσα σε ένα τετράγωνο, που έχει φύλλο, το τρώει
- Διαφορετικά, αν μπορεί να κινηθεί, μετακινείται μια θέση εμπρός
- Αν δεν μπορεί να μετακινηθεί, διότι έφτασε σε κάποιο όριο του κόσμου, τότε γυρίζει δεξιά.

Με λίγα λόγια η μέθοδος ACT (δράση) εκτελεί πολλές επιμέρους μεθόδους (τις δοκιμάσατε προηγουμένως χειροκίνητα μια μια) με μια πράξη.

```
1. public void act()
2. {
3.     if (foundLeaf()) {
4.         eatLeaf();
5.     }
6.     else if (canMove()) {
7.         move();
8.     }
9.     else {
10.        turnLeft();
11.    }
```

Το κουμπί RUN (εκκίνηση) εκτελεί ξανά και ξανά την ACT (δράση) μέχρι να το κάνετε παύση. Το κουμπί RESET (επαναφορά) επαναφέρει την οθόνη στην αρχική κατάσταση.

Το σημαντικό, όπως ίσως φανταστήκατε, είναι πίσω από όλα αυτά τα χρωματιστά αντικείμενα.



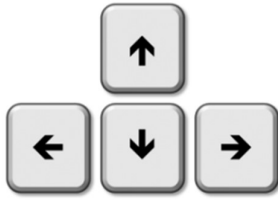
Στην οθόνη σας υπάρχει ένα πρόγραμμα γραμμένο στη γλώσσα Java. Για να δείτε το πρόγραμμα του μικρού Wombat, κάντε δεξί κλικ επάνω στην κλάση και επιλέξτε «Άνοιγμα επεξεργαστή». Εδώ θα βρείτε την υλοποίηση των μεθόδων – συμπεριφορών, αλλά και των ιδιοτήτων. Μοιάζει σαν έναν κλασσικό επεξεργαστή κειμένου, όπου μπορείτε να κάνετε τις όποιες αλλαγές. Η αποθήκευση και διερμηνευση του προγράμματος γίνεται αυτόματα από το Greenfoot.

Προσοχή! Κάθε φορά που κάνετε μια αλλαγή μέσα στο πρόγραμμα, ο κόσμος μηδενίζεται, δηλαδή γίνεται επαναφορά. Για μην βάζετε χειροκίνητα φυλλαράκια, μπορείτε να κάνετε δεξί κλικ επάνω στον κόσμο και να επιλέξετε τη μέθοδο «void randomLeaves(int howMany)». Δώστε έναν αριθμό φύλλων και αυτά θα εμφανιστούν τυχαία μέσα στον κόσμο.

Κάντε λοιπόν μια μικρή αλλαγή. Αλλάξτε τη μέθοδο void move() από move(1) σε move(4), όπως φαίνεται παρακάτω. Τοποθετήστε εκ νέου ένα Wombat και μερικά φύλλα επάνω στον νέο κόσμο. Κάντε δεξί κλικ και επιλέξτε τη συγκεκριμένη συμπεριφορά. Δείτε τη διαφορά.

```
1. /**
2.  * Move one step forward.
3.  */
4. public void move()
5. {
6.     move(4);
```

Επαναφέρετε την τιμή 1 στη μέθοδο move(). Τώρα θα προσπαθήσουμε να κάνουμε το παιχνίδι πιο προσιτό για τον απλό gamer. Στο τέλος του οδηγού αυτού υπάρχει μια λίστα συγκεκριμένων μεθόδων, που μπορείτε να χρησιμοποιήσετε. Περισσότερες πληροφορίες μπορείτε να αναζητήσετε στην online βοήθεια στη διεύθυνση <https://www.greenfoot.org/files/javadoc/>.



Θα επικεντρώσουμε την προσοχή μας στη μέθοδο `isKeyDown`, που ελέγχει αν ο χρήστης πάτησε κάποιο από τα πλήκτρα κατεύθυνσης του πληκτρολογίου, όπως το επάνω βελάκι (up), το κάτω (down), το αριστερό (left) και το δεξί (right). Για παράδειγμα, αν επιθυμείτε να ελέγξετε αν ο χρήστης πάτησε το κάτω βελάκι, τότε θα πρέπει να πληκτρολογήσετε τα εξής.

```
1. if (Greenfoot.isKeyDown("down")) {  
2.  
3.     /* Εδώ βάζετε τις εντολές εφόσον ο χρήστης πάτησε το κάτω βελάκι */  
4.  
5. }
```

Αν αντί για το down, βάλετε το up ή το left ή το right, τότε έχετε τον πλήρη έλεγχο των πλήκτρων κατεύθυνσης. Η επιδίωξη είναι, όταν ο χρήστης πατάει το συγκεκριμένο πλήκτρο, ο ήρωας να αλλάζει κατεύθυνση, δηλαδή να δείχνει προς τα κάτω, να κάνει ένα μικρό βήμα προς αυτήν την κατεύθυνση, αν μπορεί να κινηθεί (που σημαίνει ότι δεν έχει φτάσει στο όριο του κόσμου) και να τρώει το φυλλαράκι, εφόσον υπάρχει κάποιο. Δείτε τον έλεγχο αυτόν ολοκληρωμένο.

```
1. if (Greenfoot.isKeyDown("down")) {  
2.  
3.     setDirection(1); /* άλλαξε κατεύθυνση */  
4.  
5.     if (canMove()) { /* μπορεί να κινηθεί το wombat ? */  
6.         move(); /* αν ναι, τότε μετακίνησε το */  
7.     }  
8.  
9.     if (foundLeaf()) { /* το συγκεκριμένο τετράγωνο έχει κάποιο φύλλο ? */  
10.        eatLeaf(); /* αν ναι, τότε μπορείς να το φας :- ) */  
11.    }  
12. }
```

Ακριβώς το ίδιο πράγμα πρέπει να κάνετε και για τα υπόλοιπα τρία πλήκτρα κατεύθυνσης. Αντιγράψτε τον κώδικα παρακάτω και επικολλήστε τον μέσα στον κειμενογράφο της κλάσης Wombat, αντικαθιστώντας στην αρχική μέθοδο `public void act()`.

Κλείστε τον επεξεργαστή κειμένου και τοποθετήστε εκ νέου το Wombat και μερικά φυλλαράκια. Κάντε εκτέλεση και αναλάβετε τον έλεγχο του παιχνιδιού.

```

1.  /**
2.   * Do whatever the wombat likes to to just now.
3.   */
4.   public void act()
5.   {
6.       if (Greenfoot.isKeyDown("left")) {
7.           setDirection(2);
8.           if (canMove()) {
9.               move();
10.          }
11.          if (foundLeaf()) {
12.              eatLeaf();
13.          }
14.      } else if (Greenfoot.isKeyDown("right")) {
15.          setDirection(0);
16.          if (canMove()) {
17.              move();
18.          }
19.          if (foundLeaf()) {
20.              eatLeaf();
21.          }
22.      } else if (Greenfoot.isKeyDown("up")) {
23.          setDirection(3);
24.          if (canMove()) {
25.              move();
26.          }
27.          if (foundLeaf()) {
28.              eatLeaf();
29.          }
30.      } else if (Greenfoot.isKeyDown("down")) {
31.          setDirection(1);
32.          if (canMove()) {
33.              move();
34.          }
35.          if (foundLeaf()) {
36.              eatLeaf();
37.          }
38.      }
39.
40.
41.  }

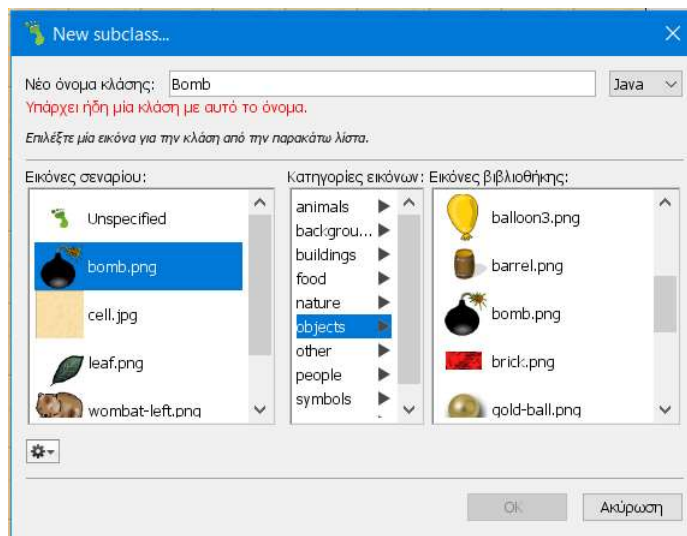
```

Κάθε παιχνίδι, που σέβεται τον εαυτό του, πρέπει να δείχνει ένα σκορ. Από τις μεθόδους που είδατε, η «void eatLeaf()» είναι αυτή, η οποία αυξάνει τον αριθμό των φύλλων που τρώει το wombat. Αυτό θα είναι και το σκορ. Μέσα σε αυτήν την μέθοδο προσθέστε τη γραμμή 10, που βλέπετε παρακάτω.

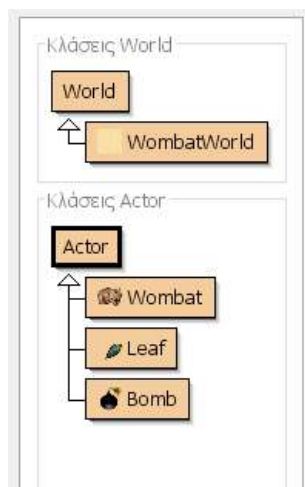
```

1.  * Eat a leaf (if there is one in our cell).
2.  */
3.  public void eatLeaf()
4.  {
5.      Actor leaf = getObjectAtOffset(0, 0, Leaf.class);
6.      if (leaf != null) {
7.          // eat the leaf...
8.          getWorld().removeObject(leaf);
9.          leavesEaten = leavesEaten + 1;
10.         getWorld().showText("Score :" + leavesEaten, 1,00);
11.     }
12. }

```



Τώρα θα πρέπει να δημιουργηθεί ο εχθρός. Κάντε δεξί κλικ επάνω στο Actor και επιλέξτε «New subclass». Στο παράθυρο, που θα ανοίξει, δώστε ως νέο όνομα κλάσης τη λέξη «Bomb», όπως το βλέπετε με το B κεφαλαίο. Στη συνέχεια, επιλέξτε την εικόνα, που θα βρείτε στην κατηγορία objects. Κάντε OK. Θα πρέπει πλέον να βλέπετε τρεις ηθοποιούς, όπως στην εικόνα παρακάτω.



Ανοίξτε τον κώδικα του κόσμου και εντοπίστε τη μέθοδο «void randomLeaves(int howMany)». Προσθέστε τις γραμμές 22 και 23 του κώδικα, που θα βρείτε παρακάτω.

```

13.     public void randomLeaves(int howMany)
14.     {
15.         for (int i=0; i<howMany; i++) {
16.             Leaf leaf = new Leaf();
17.             int x = Greenfoot.getRandomNumber(getWidth());
18.             int y = Greenfoot.getRandomNumber(getHeight());
19.             addObject(leaf, x, y);
20.         }
21.     }

```



```

22. Bomb bomb = new Bomb();
23. addObject(bomb, 1, 1);
24.
25.
26. }

```

Στη συνέχεια, ανοίξτε τον κώδικα του νέου αντικειμένου, που μόλις δημιουργήσατε, δηλαδή του Bomb και αντικαταστήστε τη μέθοδο Act με αυτή, που βλέπετε παρακάτω.

```

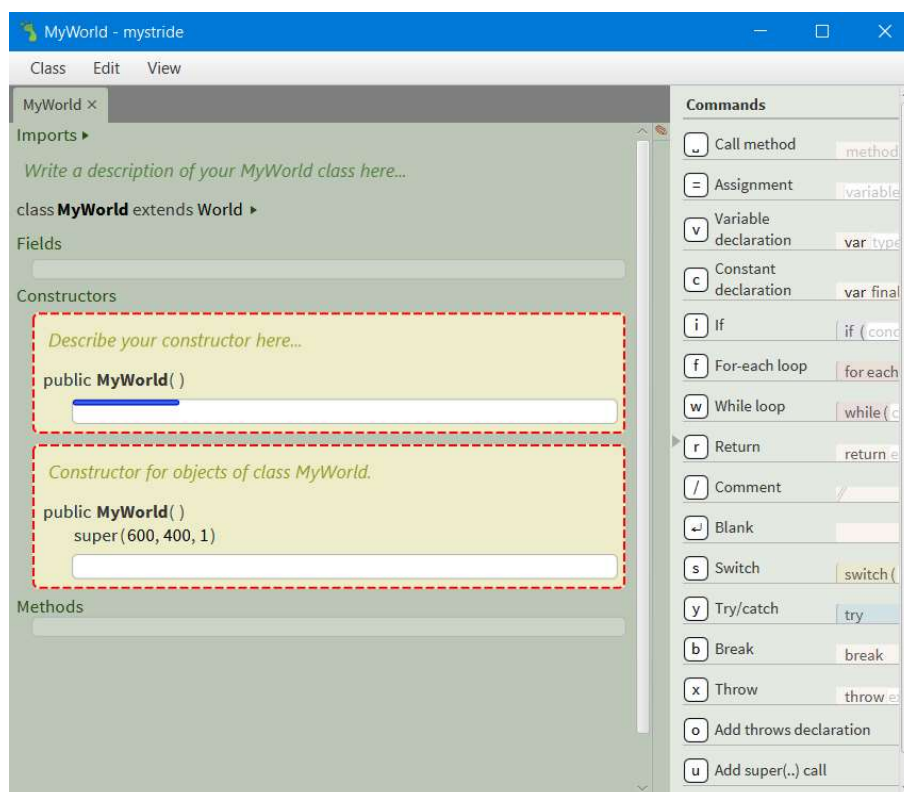
1. public class Bomb extends Actor
2. {
3.     /**
4.      * Act - do whatever the Bomb wants to do. This method is called whenever
5.      * the 'Act' or 'Run' button gets pressed in the environment.
6.      */
7.     public void act()
8.     {
9.         if ( Greenfoot.getRandomNumber(10) == 9 ){
10.             turn(90);
11.         }
12.         move(1);
13.
14.         if (isTouching(Wombat.class)){
15.             removeTouching(Wombat.class);
16.             getWorld().showText("Game over", 1,00);
17.         }
18.     }
19. }

```

Το παιχνίδι ολοκληρώθηκε. Κάντε δεξί κλικ επάνω στον κόσμο και δημιουργήστε τυχαία μερικά φύλλα. Θα εμφανιστεί επιπλέον και μια βόμβα. Κάντε δεξί κλικ επάνω στο αντικείμενο Wombat και επιλέξτε «new Wombat()». Κάντε κλικ επάνω σε ένα τετραγωνάκι του κόσμου. Κάντε εκτέλεση και δείτε το αποτέλεσμα. Η βόμβα κινείται με τυχαίο τρόπο. Αν σας πετύχει, τότε εμφανίζεται το μήνυμα «Game Over». Οι καινούργιες μέθοδοι, που προσθέσαμε, ήταν η getRandomNumber(x), η isTouching(class) και η removeTouching(class). Η πρώτη δημιουργεί έναν τυχαίο αριθμό μέχρι το x, δηλαδή από το 0 μέχρι το x-1. Η δεύτερη ελέγχει αν ο ηθοποιός αγγίζει τον ηθοποιό, που υπάρχει μέσα στην παρένθεση και η τρίτη διαγράφει τον ηθοποιό, που υπάρχει μέσα στην παρένθεση.

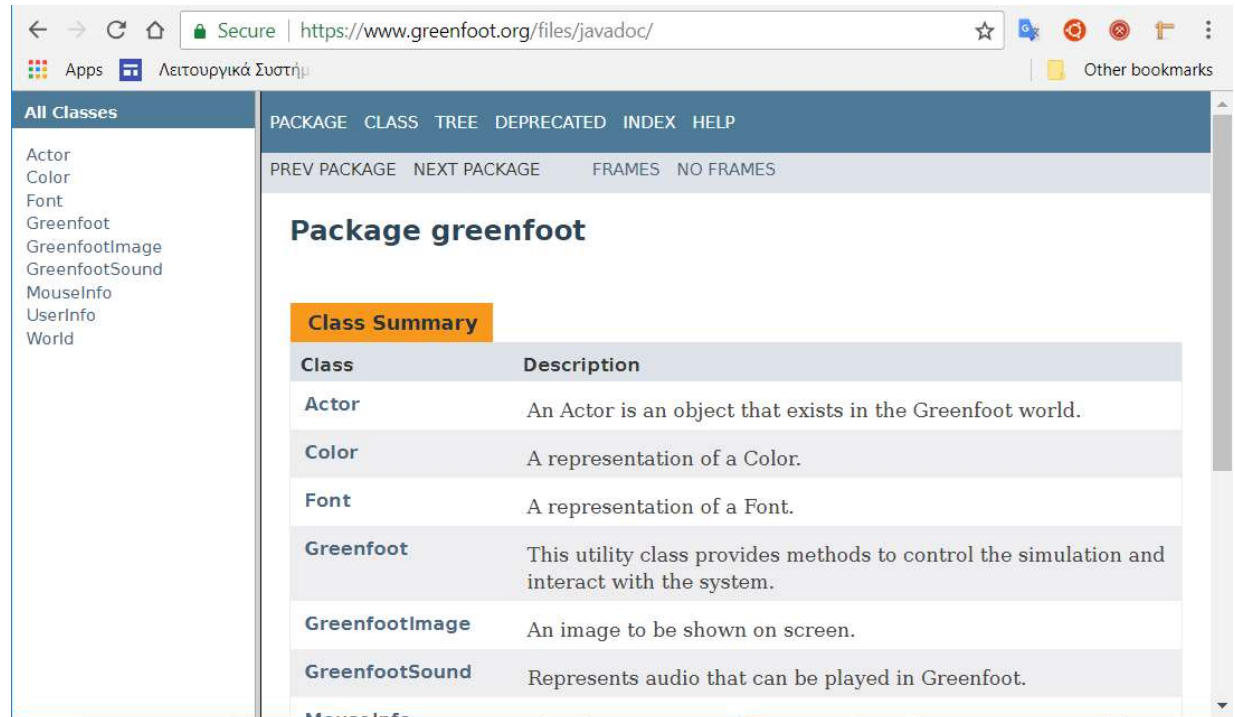
Γλώσσα Stride και επεξεργαστής κώδικα με μπλοκ (Frame Based Editor)

Η γλώσσα **Stride** είναι μια γλώσσα, που φτιάχτηκε ειδικά για το Greenfoot. Επιτρέπει την καλύτερη διαχείριση του κώδικα με τη χρήση μπλοκ, τα οποία σύρονται. Με αυτόν τον τρόπο σας διευκολύνει στη συγγραφή του κώδικα με έτοιμα μπλοκ, τα οποία μπορείτε να βρείτε ακριβώς δεξιά του επεξεργαστή κειμένου. Όταν τελειώσετε με τον κώδικα το Greenfoot, θα μετατρέψει τον κώδικα Stride σε κώδικα Java. Μπορείτε να δείτε πως λειτουργεί, αν ξεκινήσετε ένα νέο stride scenario στην επιλογή των μενού «Σενάριο».



Βασικές συναρτήσεις για τον χειρισμό και τον έλεγχο των αντικειμένων στην οθόνη
Μπορείτε να έχετε πλήρη πρόσβαση σε όλες τις συναρτήσεις μέσω της διεύθυνσης

► <https://www.greenfoot.org/files/javadoc/>



public void move(int distance)

Μετακινεί τον ηθοποιό σε απόσταση, που ορίζεται από το distance προς την κατεύθυνση, που αυτός «βλέπει».

Παράδειγμα : move (4)

public void turn(int amount)

Περιστρέφει τον ηθοποιό τόσες μοίρες, όσες ορίζονται στην παράμετρο degrees.

Παράδειγμα : turn(90)

protected boolean intersects(Actor other)

Ελέγχει αν ο ηθοποιός «συγκρούεται» με κάποιον άλλο ηθοποιό στην οθόνη. Το αποτέλεσμα είναι true, αν ισχύει κάτι τέτοιο ή false, αν δεν ισχύει.

Παράδειγμα : if (intersects(ball))

public boolean isAtEdge()

Ελέγχει αν ο ηθοποιός έχει φτάσει στα όρια του κόσμου (world) μέσα στην οθόνη για μια συγκεκριμένη ανάλυση. Αν ο κόσμος έχει οριζοντίως 640 εικονοστοιχεία, τότε τα όρια σε αυτήν την περίπτωση είναι η τιμή 0 ή 640. Επιστρέφει true, αν ισχύει κάτι τέτοιο, διαφορετικά επιστρέφει false.

Παράδειγμα : if (isAtEdge())

protected boolean isTouching(java.lang.Class<?> cls)

Ελέγχει αν ο ηθοποιός «αγγίζει» κάποια άλλα αντικείμενα μιας δοθείσας κλάσης. Επιστρέφει true αν ισχύει κάτι τέτοιο, διαφορετικά επιστρέφει false.

Παράδειγμα : If (isTouching("Cherry.class"))

protected void removeTouching(java.lang.Class<?> cls)

Διαγράφει κάποια από τα αντικείμενα μια κλάσης αν ο ηθοποιός τα αγγίζει. Για παράδειγμα διαγράφει τις τελείες στο γνωστό παιχνίδι pacman, όταν ο κεντρικός χαρακτήρας και ηθοποιός, στην περίπτωση του Greenfoot, τις αγγίζει.

Παράδειγμα : removeTouching("Cherry.class")

public static int getRandomNumber(int limit)

Επιστρέφει έναν τυχαίο αριθμό από το 0 έως το όριο limit (δηλαδή 0 έως limit-1), που θα οριστεί.

Παράδειγμα : getRandomNumber(100)

public void showText(java.lang.String text, int x, int y)

Εμφανίζει το επιλεγμένο κείμενο text στη θέση x και y επάνω στην οθόνη του κόσμου (world). Οι ηθοποιοί βρίσκονται ένα επίπεδο πιο κάτω από το κείμενο και έτσι ποτέ δεν διαγράφεται μερικώς ή ολικώς.

Παράδειγμα : getWorld().showText("Score", 300, 200)

public static boolean isKeyDown(java.lang.String keyName)

Ελέγχει αν ο χρήστης πάτησε κάποιο πλήκτρο από το πληκτρολόγιο του. Για τα πλήκτρα κατεύθυνσης ισχύουν οι εξής ονομασίες : left (αριστερό), right (δεξί), up (επάνω) και down (κάτω).

Παράδειγμα : if (Greenfoot.isKeyDown("left"))

ΠΑΡΑΡΤΗΜΑ II – ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ

Λάρισα, Οκτώβριος 2018

Η παρούσα έρευνα διεξάγεται στο πλαίσιο εκπόνησης διπλωματικής εργασίας για την απόκτηση μεταπτυχιακού τίτλου σπουδών στο τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική του Πανεπιστημίου Θεσσαλίας. Στόχος της έρευνας είναι να αναδείξει την αποτελεσματικότητα ή μη της χρήσης παιχνιδιών για την κατανόηση και εμπέδωση του Προγραμματισμού.

Η διεξαγωγή της έρευνας είναι δυνατή μόνο με τη δική σας συμβολή, που αφορά στη συμπλήρωση ερωτηματολογίου.

Παρακαλώ να απαντήσετε ατομικά στις ερωτήσεις, που ακολουθούν. Η απάντηση δίνεται, είτε σημειώνοντας Χ στο τετραγωνάκι, που βρίσκεται μπροστά από κάθε απάντηση, είτε γράφοντας την απάντηση, όπου δεν υπάρχει τετραγωνάκι, αλλά διακεκομμένη γραμμή. Όπου απαιτείται, υπάρχουν επιπλέον οδηγίες.

Σας ενημερώνω ότι το ερωτηματολόγιο είναι ανώνυμο και τα αποτελέσματα θα χρησιμοποιηθούν αυστηρά και μόνο στη στατιστική ανάλυση της έρευνας.

Σας ευχαριστώ εκ των προτέρων για τη συμμετοχή και τον χρόνο σας.

Με εκτίμηση

Μόχλα Αναστασία

Μεταπτυχιακή Φοιτήτρια του Πανεπιστημίου Θεσσαλίας (ΤΠΕ στην Εκπαίδευση)

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ

Το ερωτηματολόγιο είναι ανώνυμο. Σημειώστε X, όπου υπάρχει ☐ και συμπληρώστε με κείμενο, όπου υπάρχει διακεκομμένη γραμμή.

Φύλο: Γυναίκα ☐ Άνδρας ☐

Ηλικία:

☐ 18....22

☐ 23....30

☐ 31....40

☐ 41....70

Ερωτήσεις

A. Εμπειρία στη χρήση Η/Υ, τον προγραμματισμό και άλλα περιβάλλοντα

1. Αξιολογήστε τις γνώσεις σας στη χρήση Η/Υ.

(Επιλέξτε μία απάντηση, όπου 1 =Λίγες και 5=Άριστες)

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5

2. Αξιολογήστε τις γνώσεις στον προγραμματισμό Η/Υ.

(Επιλέξτε μία απάντηση, όπου 1 =Λίγες και 5=Άριστες)

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5

3. Σημειώστε με ποια λογισμικά έχετε ασχοληθεί στο παρελθόν.

(Μπορείτε να επιλέξετε περισσότερες από μία απαντήσεις)

- ☐ Greenfoot
- ☐ Alice
- ☐ Scratch
- ☐ MicroWorlds
- ☐ Lego MindStorms
- ☐ Codu
- ☐ Stencyl

Άλλο:

4. Σημειώστε τις γλώσσες προγραμματισμού, με τις οποίες έχετε ασχοληθεί στο παρελθόν.
(Μπορείτε να επιλέξετε περισσότερες από μία απαντήσεις)

- ☐ Pascal (οποιαδήποτε έκδοση πχ DOS, Windows, Linux κτλ)
- ☐ C/C++
- ☐ Java
- ☐ Assembly (οποιοδήποτε επεξεργαστή πχ 6502, 68000, x86 κτλ)
- ☐ Basic
- ☐ Python
- ☐ PHP

Άλλο:.....

5. Χρησιμοποιείτε γενικά κάποια γλώσσα προγραμματισμού για τη δουλειά σας αυτό το διάστημα;

- ☐ Ναι
- ☐ Όχι

B. Κατανόηση διδακτικού αντικειμένου

6. Σε τι διαφέρει ο αντικειμενοστραφής προγραμματισμός από τον διαδικαστικό/συναρτησιακό;

(Μπορείτε να επιλέξετε περισσότερες από μία απαντήσεις.)

- ☐ Δεν χρησιμοποιεί συναρτήσεις αλλά μεθόδους
- ☐ Δεν υπάρχει καμία απολύτως διαφορά
- ☐ Είναι πιο εύκολη η ανάλυση του προβλήματος, άρα και η υλοποίησή του
- ☐ Η περιγραφή του προβλήματος μέσω του προγράμματος γίνεται με τους όρους του προβλήματος και όχι της μηχανής

7. Τι είναι η κληρονομικότητα των κλάσεων;

- ☐ Τα νέα στιγμιότυπα των κλάσεων
- ☐ Η δημιουργία των ιδιοτήτων και μεθόδων μέσα σε μια κλάση
- ☐ Η δυνατότητα των νέων κλάσεων να κληρονομούν τις ιδιότητες άλλων κλάσεων

8. Πώς εκφράζονται οι συμπεριφορές μέσα σε ένα αντικείμενο;

- ☐ Μέσω των μεθόδων
- ☐ Μέσω των ιδιοτήτων
- ☐ Μέσω της κληρονομικότητας

9. Τι είναι το στιγμιότυπο μιας κλάσης;

- ☐ Η δημιουργία ενός αντικειμένου, που ανήκει σε μια κλάση, είναι στιγμιότυπο (instance) αυτής της κλάσης
- ☐ Είναι οι κλάσεις που κληρονομούν άλλες κλάσεις
- ☐ Η ιδιότητα που προσφέρουν οι κλάσεις να «κρύβουν» τα ιδιωτικά δεδομένα τους από το υπόλοιπο πρόγραμμα

10. Το στιγμιότυπο μιας κλάσης μπορεί να έχει διαφορετικά δεδομένα στις ιδιότητές του από κάποιο άλλο της ίδιας κλάσης;

☐ Ναι

☐ Όχι

Γ. Το περιβάλλον του Greenfoot.

11. Πώς σας φάνηκε το περιβάλλον ανάπτυξης του GreenFoot;

☐ Εύκολο

☐ Εύκολο και Διασκεδαστικό

☐ Δύσκολο

☐ Δύσχρηστο

12. Τι βαθμό δυσκολίας θα ορίζατε για τη χρήση του περιβάλλοντος ανάπτυξης του Greenfoot;
(Επιλέξτε μία απάντηση, όπου 1 = Πολύ εύκολο και 5 = Πολύ δύσκολο)

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

13. Το προτιμάτε από ένα συμβατικό περιβάλλον ανάπτυξης μιας γλώσσας προγραμματισμού;

☐ Ναι

☐ Όχι

14. Προτιμάτε τον επεξεργαστή κειμένου της γλώσσας Java ή της γλώσσας Stride με τη μέθοδο του frame based editor;

☐ Java

☐ Stride

15. Θα το συνιστούσατε ως εκπαιδευτικό εργαλείο για την εκμάθηση της γλώσσας προγραμματισμού Java;

☐ Ναι

☐ Όχι

Ευχαριστώ για τη συμμετοχή

ΠΑΡΑΡΤΗΜΑ ΙΙΙ – ΕΙΚΟΝΕΣ

Εικόνα 1. Το προγραμματιστικό περιβάλλον Alice	29
Εικόνα 2. Οι ηλικιακές ομάδες, στις οποίες απευθύνονται τα περιβάλλοντα Alice, Scratch και Greenfoot (Woei, L.S., IH Othman I. H. & Man C. K. (2013))	29
Εικόνα 3. Το περιβάλλον του Greenfoot	34
Εικόνα 4. Η αρχή και το τέλος των πλαισίων της Stride	36
Εικόνα 5. Παράδειγμα υλοποίησης μιας απλής ACT μεθόδου (μετακινεί τον actor - μπάλα ποδοσφαίρου - τέσσερα βήματα αριστερά από την τρέχουσα θέση του και ταυτόχρονα τον περιστρέφει κατά δύο μοίρες επίσης αριστερά)	38
Εικόνα 6. Η αρχική σελίδα του ιστότοπου Greenfoot (www.greenfoot.org)	39
Εικόνα 7. Το παράθυρο και οι βασικές εργαλειοθήκες στο περιβάλλον Greenfoot	41
Εικόνα 8. Κονσόλα χειρισμού	42
Εικόνα 9. Άνοιγμα αρχείου και αποθήκευση με άλλο όνομα	43
Εικόνα 10. Το αρχικό παράθυρο	43
Εικόνα 11. Εμφάνιση του «κόσμου» μετά τη μεταγλώττιση	44
Εικόνα 12. Εισαγωγή αντικειμένου	44
Εικόνα 13. Διαγραφή αντικειμένου	45
Εικόνα 14. Η λειτουργία «Act»	45
Εικόνα 15. Μέθοδοι, που αφορούν τον «Actor»	46
Εικόνα 16. Η μέθοδος «populate()»	47
Εικόνα 17. Η μέθοδος «randomLeaves(int howMany)»	47
Εικόνα 18. Open Editor	48
Εικόνα 19. Ο κώδικας της μεθόδου «randomLeaves(int howMany)»	49
Εικόνα 20. Η εντολή «super()» και η μέθοδος «setBackground()»	49
Εικόνα 21. Αλλαγή αρχείου με χρήση κώδικα	50
Εικόνα 22. Ο νέος "Κόσμος"	50
Εικόνα 23. Δίνοντας κίνηση στα «φύλλα»	51
Εικόνα 24. Οι μέθοδοι «eatLeaf()», «move()», και «turnLeft()»	52
Εικόνα 25. Οι γενικές κατευθύνσεις	52
Εικόνα 26. Χρήση πλήκτρων κατεύθυνσης	53
Εικόνα 27. Αλλαγή κατεύθυνσης (direction)	53
Εικόνα 28. Οι μέθοδοι «Counter()», «UpdateImage()» και «addScore()»	54
Εικόνα 29. Μέσα στις μεθόδους του «wombat» προστέθηκαν τέσσερις νέες εντολές	55
Εικόνα 30. Εμφάνιση του «score»	56

ΠΑΡΑΡΤΗΜΑ IV – ΠΙΝΑΚΕΣ

Πίνακας 1. Συνοπτικός πίνακας μελετών σχετικά με το εργαλείο Greenfoot (Τατόγλου, 2018).....	32
Πίνακας 2. Φύλο δείγματος.....	62
Πίνακας 3. Ηλικία δείγματος	63
Πίνακας 4. Γνώσεις Η/Υ.....	64
Πίνακας 5. Γνώσεις σε προγραμματισμό	64
Πίνακας 6. Λογισμικά (πρότερη γνώση).....	65
Πίνακας 7. Γλώσσες προγραμματισμού (πρότερη γνώση).....	65
Πίνακας 8. Χρήση γλώσσας προγραμματισμού στην εργασία.....	66
Πίνακας 9. Διαφορές αντικειμενοστραφούς και διαδικαστικού/συναρτησιακού προγραμματισμού	67
Πίνακας 10. Κληρονομικότητα κλάσεων.....	67
Πίνακας 11. Τρόπος έκφρασης συμπεριφορών μέσα σε αντικείμενο	68
Πίνακας 12. Στιγμιότυπο κλάσης.....	68
Πίνακας 13. Ιδιότητες στιγμιότυπου κλάσης	69
Πίνακας 14. Άποψη για το περιβάλλον Greenfoot	70
Πίνακας 15. Βαθμός δυσκολίας.....	70
Πίνακας 16. Greenfoot ή συμβατικό περιβάλλον προγραμματισμού;	71
Πίνακας 17. Επεξεργαστής κειμένου Java ή Stride;	72
Πίνακας 18. Σύσταση Greenfoot ως εκπαιδευτικό εργαλείο εκμάθησης Java	72